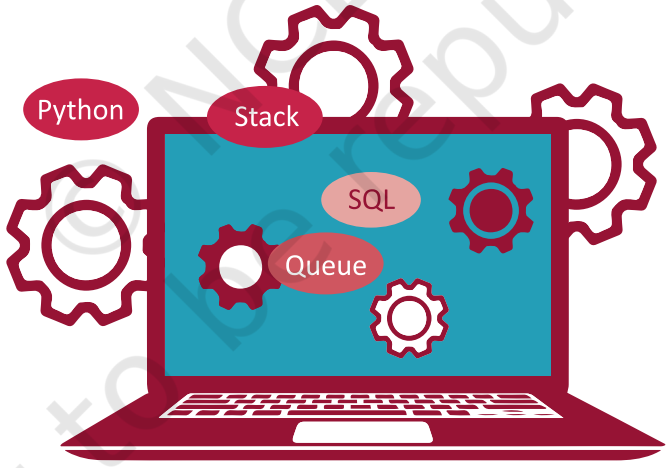
Computer Science

Textbook for Class XII

12130



© NCERT not to be republished

विद्यया ऽमृतमश्नुते
एन सी ई आर टी
NCERT

राष्ट्रीय शैक्षिक अनुसंधान और प्रशिक्षण परिषद्
NATIONAL COUNCIL OF EDUCATIONAL RESEARCH AND TRAINING

Reprint 2026-27

12130 – COMPUTER SCIENCE
Textbook for Class XII

ISBN 978-93-5292-338-0

First Edition
September 2020 Bhadrapada 1942

Reprinted
September 2021 Bhadrapada 1943
December 2021 Agrahayana 1943
October 2022 Kartika 1944
March 2024 Chaitra 1946
June 2024 Jyeshtha 1946
December 2025 Agrahayana 1947

PD 45T HK

© *National Council of Educational Research and Training, 2020*

₹ 250.00

Printed on 80 GSM paper with NCERT watermark

Published at the Publication Division by the Secretary, National Council of Educational Research and Training, Sri Aurobindo Marg, New Delhi 110 016 and printed at Chandra Prabhu Offset Printing Works (P) Ltd., C-40, Sector-8, Noida 201 301 (UP)

ALL RIGHTS RESERVED

- No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission of the publisher.
- This book is sold subject to the condition that it shall not, by way of trade, be lent, re-sold, hired out or otherwise disposed off without the publisher's consent, in any form of binding or cover other than that in which it is published.
- The correct price of this publication is the price printed on this page. Any revised price indicated by a rubber stamp or by a sticker or by any other means is incorrect and should be unacceptable.

OFFICES OF THE PUBLICATION DIVISION, NCERT

NCERT Campus
Sri Aurobindo Marg
New Delhi 110 016 **Phone : 011-26562708**

108, 100 Feet Road
Hosdakere Halli Extension
Banashankari III Stage
Bengaluru 560 085 **Phone : 080-26725740**

Navjivan Trust Building
P.O.Navjivan
Ahmedabad 380 014 **Phone : 079-27541446**

CWC Campus
Opp. Dhankal Bus Stop
Panihati
Kolkata 700 114 **Phone : 033-25530454**

CWC Complex
Maligaon
Guwahati 781 021 **Phone : 0361-2674869**

Publication Team

Head, Publication Division	:	*M.V. Srinivasan*
Chief Editor	:	*Bijnan Sutar*
Chief Business Manager	:	*Amitabh Kumar*
Chief Production Officer (In charge)	:	*Deepak Jaiswal*
Assistant Production Officer	:	*Deepak Kumar*

Cover and Layout
Meetu Sharma, DTP Operator, DESM

© NCERT not to be republished

Foreword

Computer science as a discipline has evolved over the years and has emerged as a driving force of our socio-economic activities. It has made continuous inroads into diverse areas — be it business, commerce, science, technology, sports, health, transportation or education. With the advent of computer and communication technologies, there has been a paradigm shift in teaching-learning at the school level. The role and relevance of this discipline is in focus because the expectations from the school pass-outs have grown to be able to meet the challenges of the 21st century. Today, we are living in an interconnected world where computer-based applications influence the way we learn, communicate, commute or even socialise!

There is a demand for software engineers in various fields like manufacturing, services, etc. Today, there are a large number of successful startups delivering different services through software applications. All these have resulted in generating interest for this subject among students as well as parents.

Development of logical thinking, reasoning and problem-solving skills are fundamental building blocks for knowledge acquisition at the higher level. Computer plays a key role in problem solving with focus on logical representation or reasoning and analysis.

This textbook focuses on the fundamental concepts and problem-solving skills while opening a window to the emerging and advanced areas of computer science. The newly developed syllabus has dealt with the dual challenge of reducing curricular load as well as introducing this ever evolving discipline. This textbook also provides space to Computational Thinking and Artificial Intelligence, which envisaged in National Education Policy, 2020.

As an organisation committed to systemic reforms and continuous improvement in the quality of its products, NCERT welcomes comments and suggestions which will enable us to revise the content of the textbook.

Hrushikesh Senapaty
Director
National Council of Educational
Research and Training

New Delhi
August 2020

© NCERT not to be republished

© NCERT
not to be republished

Preface

In the present education system of our country, specialised or discipline based courses are introduced at the higher secondary stage. This stage is crucial as well as challenging because of the transition from general to discipline-based curriculum. The syllabus at this stage needs to have sufficient rigour and depth while remaining mindful of the comprehension level of the learners. Further, the textbook should not be heavily loaded with content.

Computers have permeated in every facet of life. Study of basic concepts of computer science has been desirable in education. There are courses offered in the name of Computer Science, Information and Communication Technology (ICT), Information Technology (IT), etc., by various boards and schools up to secondary stage, as optional. These mainly focus on using computer for word processing, presentation tools and application software.

Computer Science (CS) at the higher secondary stage of school education is also offered as an optional subject. At this stage, students usually opt for CS with an aim of pursuing a career in software development or related areas, after going through professional courses at higher levels. Therefore, at higher secondary stage, the curriculum of CS introduces basics of computing and sufficient conceptual background of Computer Science.

The primary focus is on fostering the development of computational thinking and problem-solving skills. This book has 13 chapters covering the following broader themes:

- Data Structure: understanding of important data structure Stack, Queue; Searching and Sorting techniques.
- Database: basic understanding of data, database concepts, and relational database management system using MySQL. Structured query language—data definition, data manipulation and data querying.
- Programming: handling errors and exceptions in programs written in Python; handling files and performing file operations in Python.
- Network and Communication: fundamentals of Computers networks, devices, topologies, Internet, Web and IoT, DNS. Basics of Data communication—transmission channel, media; basics of protocols, mobile communication generations.
- Security Aspects: introduction to basic concepts related to network and Internet security, threats and prevention.

Each chapter has two additional components—(i) activities and (ii) think and reflect for self assessment while learning as well as to generate further interest in the learner. A number of hands-on examples are given to gradually explain methodology to solve different types of problems across the Chapters. The programming examples as well as the exercises in the

© NCERT not to be republished

chapters are required to be solved in a computer and verify with the given outputs.

Box items are pinned inside the chapters either to explain related concepts or to describe additional information related to the topic covered in that section. However, these box-items are not to be assessed through examinations.

Project Based Learning given as the end includes exemplar projects related to real-world problems. Teachers are supposed to assign these or similar projects to be developed in groups. Working in such projects may promote peer-learning, team spirit and responsiveness.

The chapters have been written by involving practicing teachers as well as subject experts. Several iterations have resulted into this book. Thanks are due to the authors and reviewers for their valuable contribution. I would like to place on record appreciation for Professor Om Vikas for leading the review activities of the book as well as for his guidance and motivation to the development team throughout. Comments and suggestions are welcome.

New Delhi
20 August 2020

Rejaul Karim Barbhuiya
Assistant Professor
Central Institute of
Educational Technology

© NCERT not to be republished

Textbook Development Committee

Chief Advisor

Om Vikas, *Professor* (Retd.), Former Director, ABV-IIITM, Gwalior, M.P.

Members

Anju Gupta, *Freelance Educationist,* Delhi

Anuradha Khattar, *Assistant Professor*, Miranda House, University of Delhi

Chetna Khanna, *Freelance Educationist*, Delhi

Faheem Masoodi, *Assistant Professor*, Department of Computer Science, University of Kashmir

Harita Ahuja, *Assistant Professor*, Acharya Narendra Dev College, University of Delhi

Mohini Arora, *HOD, Computer Science*, Air Force Golden Jubilee Institute, Subroto Park, Delhi

Mudasir Wani, *Assistant Professor*, Govt. College for Women Nawakadal, Sri Nagar, Jammu and Kashmir

Naeem Ahmad, *Assistant Professor*, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh

Purvi Kumar, *Co-ordinator*, Computer Science Department, Ganga International School, Rohtak Road, Delhi

Priti Rai Jain, *Assistant Professor*, Miranda House, University of Delhi

Sangita Chadha, *HOD, Computer Science*, Ambience Public School, Safdarjung Enclave, Delhi

Sharanjit Kaur, *Associate Professor*, Acharya Narendra Dev College, University of Delhi

Member-coordinator

Rejaul Karim Barbhuiya, *Assistant Professor*, CIET, NCERT, Delhi

© NCERT not to be republished

Acknowledgements

The National Council of Educational Research and Training acknowledges the valuable contributions of the individuals and organisations involved in the development of Computer Science textbook for Class XII.

The Council expresses its gratitude to the syllabus development team including MPS Bhatia, *Professor*, Netaji Subhas Institute of Technology, Delhi; T. V. Vijay Kumar, *Professor*, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi; Zahid Raza, *Associate Professor*, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi; Vipul Shah, *Principal Scientist*, Tata Consultancy Services, and the CSpathshala team; Aasim Zafar, *Associate Professor*, Department of Computer Science, Aligarh Muslim University, Aligarh; Faisal Anwer, *Assistant Professor*, Department of Computer Science, Aligarh Muslim University, Aligarh; Smruti Ranjan Sarangi, *Associate Professor*, Department of Computer Science and Engineering, Indian Institute of Technology, Delhi; Vikram Goyal, *Associate Professor,* Indraprastha Institute of Information Technology (IIIT), Delhi; and Mamur Ali, *Assistant Professor*, Department of Teacher Training and Non-formal Education (IASE), Faculty of Education, Jamia Millia Islamia, New Delhi.

The Council is thankful to the following resource persons for providing valuable inputs in developing this book — Veer Sain Dixit, *Assistant Professor*, Atma Ram Sanatan Dharma College, University of Delhi; Mukesh Kumar, DPS RK Puram, Delhi; Aswin K. Dash, Mother's International School, Delhi; Anamika Gupta, *Assistant Professor*, Shaheed Sukhdev College of Business Studies, University of Delhi, Sajid Yousuf Bhat, *Assistant Professor*, University of Kashmir, Jammu and Kashmir.

The council is grateful to Prof. Sunita Farkya, *Head*, Department of Education in Science and Mathematics, NCERT and Prof. Amarendra P. Behera, *Joint Director*, CIET, NCERT for their valuable cooperation and support throughout the development of this book.

The Council also gracefully acknowledges the contributions of Meetu Sharma, *Graphic Designer cum DTP Operator*; Kanika Walecha, *DTP Operator*; Pooja, *Junior Project Fellow*; in shaping this book. The contributions of the office of the APC, DESM and Publication Division, NCERT, New Delhi, in bringing out this book are also duly acknowledged.

The Council also acknowledges the contribution of Ankeeta Bezboruah *Assistant Editor* (Contractual) Publication Division, NCERT for copy editing this book. The efforts of Naresh Kumar, *DTP Operator* (Contractual), Publication Division, NCERT are also acknowledged.

© NCERT not to be republished

Contents

Foreword *iii*

Preface *v*

Chapter 1 Exception Handling in Python **1**

1.1 Introduction 1
1.2 Syntax Errors 1
1.3 Exceptions 3
1.4 Built-in Exceptions 3
1.5 Raising Exceptions 4
1.6 Handling Exceptions 7
1.7 Finally Clause 13

Chapter 2 File Handling in Python **19**

2.1 Introduction to Files 19
2.2.Types of Files 20
2.3 Opening and Closing a Text File 21
2.4 Writing to a Text File 23
2.5 Reading from a Text File 25
2.6 Setting Offsets in a File 28
2.7 Creating and Traversing a Text File 29
2.8 The Pickle Module 32

Chapter 3 Stack **39**

3.1 Introduction 39
3.2 Stack 40
3.3 Operations on Stack 42
3.4 Implementation of Stack in Python 43
3.5 Notations for Arithmetic Expressions 46
3.6 Conversion from Infix to Postfix Notation 47
3.7 Evaluation of Postfix Expression 49

Chapter 4 Queue **53**

4.1 Introduction to Queue 53
4.2 Operations on Queue 55

© NCERT not to be republished

4.3 Implementation of Queue using Python 56
4.4 Introduction to Deque 59
4.5 Implementation of Deque Using Python 61

CHAPTER 5 SORTING 67
5.1 Introduction 67
5.2 Bubble Sort 68
5.3 Selection Sort 71
5.4 Insertion Sort 74
5.5 Time Complexity of Algorithms 77

CHAPTER 6 SEARCHING 81
6.1 Introduction 81
6.2 Linear Search 82
6.3 Binary Search 85
6.4 Search by Hashing 90

CHAPTER 7 UNDERSTANDING DATA 97
7.1 Introduction to Data 97
7.2 Data Collection 101
7.3 Data Storage 102
7.4 Data Processing 102
7.5 Statistical Techniques for Data Processing 103

CHAPTER 8 DATABASE CONCEPTS 111
8.1 Introduction 111
8.2 File System 112
8.3 Database Management System 115
8.4 Relational Data Model 120
8.5 Keys in a Relational Database 123

CHAPTER 9 STRUCTURED QUERY LANGUAGE (SQL) 131
9.1 Introduction 131
9.2 Structured Query Language (SQL) 131
9.3 Data Types and Constraints in MySQL 133
9.4 SQL for Data Definition 134
9.5 SQL for Data Manipulation 141
9.6 SQL for Data Query 144
9.7 Data Updation and Deletion 154
9.8 Functions in SQL 156
9.9 GROUP BY Clause in SQL 167

© NCERT not to be republished

9.10 Operations on Relations 169
9.11 Using Two Relations in a Query 172

Chapter 10 Computer Networks 181
10.1 Introduction to Computer Networks 181
10.2 Evolution of Networking 183
10.3 Types of Networks 184
10.4 Network Devices 187
10.5 Networking Topologies 191
10.6 Identifying Nodes in a Networked Communication 194
10.7 Internet, Web and the Internet of Things 195
10.8 Domain Name System 197

Chapter 11 Data Communication 203
11.1 Concept of Communication 203
11.2 Components of data Communication 204
11.3 Measuring Capacity of Communication Media 205
11.4 Types of Data Communication 206
11.5 Switching Techniques 208
11.6 Transmission Media 209
11.7 Mobile Telecommunication Technologies 215
11.8 Protocol 216

Chapter 12 Security Aspects 223
12.1 Threats and Prevention 223
12.2 Malware 224
12.3 Antivirus 230
12.4 Spam 231
12.5 HTTP vs HTTPS 231
12.6 Firewall 232
12.7 Cookies 233
12.8 Hackers and Crackers 234
12.9 Network Security Threats 235

Chapter 13 Project Based Learning 241
13.1 Introduction 241
13.2 Approaches for Solving Projects 242
13.3 Teamwork 243
13.4 Project Descriptions 245

© NCERT not to be republished

THE CONSTITUTION OF INDIA

PREAMBLE

WE, THE PEOPLE OF INDIA, having solemnly resolved to constitute India into a [1]**[SOVEREIGN SOCIALIST SECULAR DEMOCRATIC REPUBLIC]** and to secure to all its citizens :

JUSTICE, social, economic and political;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity; and to promote among them all

FRATERNITY assuring the dignity of the individual and the [2][unity and integrity of the Nation];

IN OUR CONSTITUENT ASSEMBLY this twenty-sixth day of November, 1949 do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

---

1. Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Sovereign Democratic Republic" (w.e.f. 3.1.1977)
2. Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Unity of the Nation" (w.e.f. 3.1.1977)

Chapter 1

Exception Handling in Python

12130CH01

"I like my code to be elegant and efficient. The logic should be straightforward to make it hard for bugs to hide, the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so as not to tempt people to make the code messy with unprincipled optimization. Clean code does one thing well."

— Bjarne Stroustrup

In this Chapter

- *Introduction*
- *Syntax Errors*
- *Exceptions*
- *Built-in Exceptions*
- *Raising Exceptions*
- *Handling Exceptions*
- *Finally Clause*

1.1 Introduction

Sometimes while executing a Python program, the program does not execute at all or the program executes but generates unexpected output or behaves abnormally. These occur when there are syntax errors, runtime errors or logical errors in the code. In Python, exceptions are errors that get triggered automatically. However, exceptions can be forcefully triggered and handled through program code. In this chapter, we will learn about exception handling in Python programs.

1.2 Syntax Errors

Syntax errors are detected when we have not followed the rules of the particular programming language while writing a program. These errors are also known as *parsing errors*. On encountering a syntax error, the interpreter does not execute the program unless we rectify the errors, save and

© NCERT not to be republished

rerun the program. When a syntax error is encountered while working in shell mode, Python displays the name of the error and a small description about the error as shown in Figure 1.1.

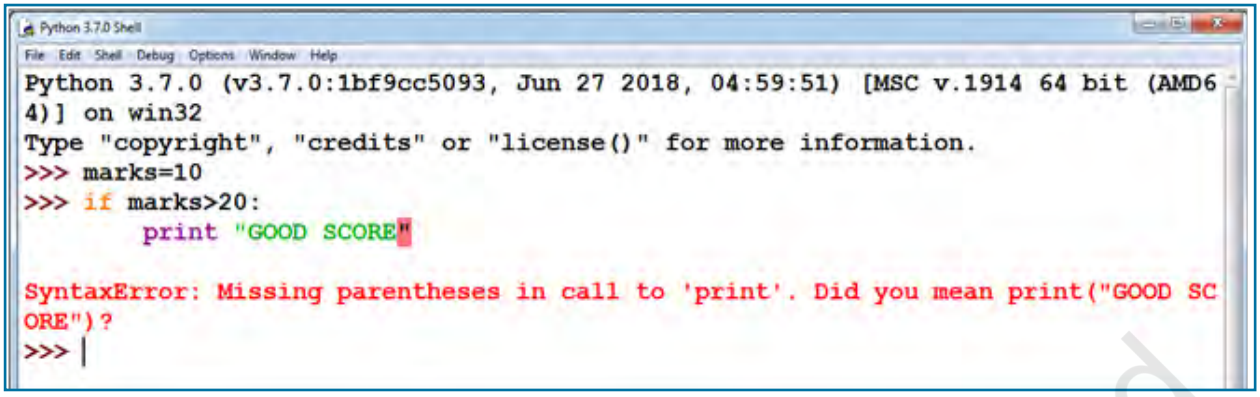


Figure 1.1: A syntax error displayed in Python shell mode

So, a syntax error is reported by the Python interpreter giving a brief explanation about the error and a suggestion to rectify it.

Similarly, when a syntax error is encountered while running a program in script mode as shown in Figure 1.2, a dialog box specifying the name of the error (Figure 1.3) and a small description about the error is displayed.

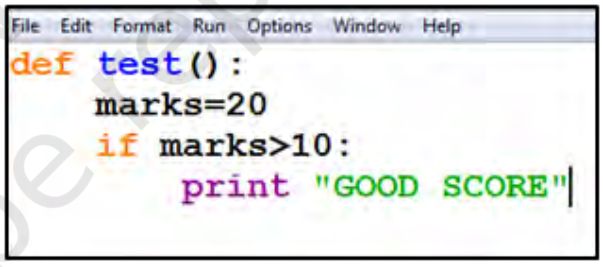


Figure 1.2: An error in the script

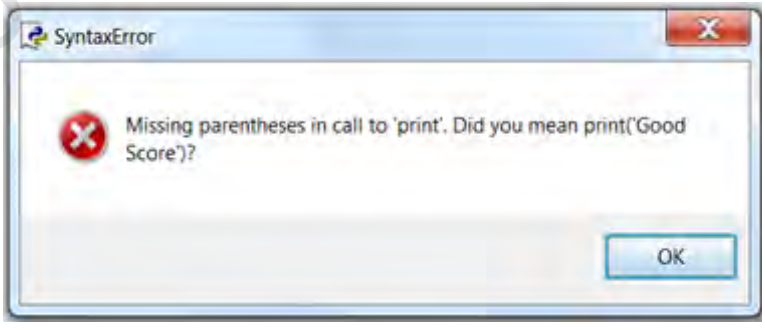


Figure 1.3: Error dialog box

© NCERT not to be republished

Reprint 2026-27

1.3 Exceptions

Even if a statement or expression is syntactically correct, there might arise an error during its execution. For example, trying to open a file that does not exist, division by zero and so on. Such types of errors might disrupt the normal execution of the program and are called exceptions.

An exception is a Python object that represents an error. When an error occurs during the execution of a program, an exception is said to have been raised. Such an exception needs to be handled by the programmer so that the program does not terminate abnormally. Therefore, while designing a program, a programmer may anticipate such erroneous situations that may arise during its execution and can address them by including appropriate code to handle that exception.

It is to be noted that SyntaxError shown at Figures 1.1 and 1.3 is also an exception. But, all other exceptions are generated when a program is syntactically correct.

1.4 Built-in Exceptions

Commonly occurring exceptions are usually defined in the compiler/interpreter. These are called built-in exceptions.

Python's standard library is an extensive collection of built-in exceptions that deals with the commonly occurring errors (exceptions) by providing the standardized solutions for such errors. On the occurrence of any built-in exception, the appropriate exception handler code is executed which displays the reason along with the raised exception name. The programmer then has to take appropriate action to handle it. Some of the commonly occurring built-in exceptions that can be raised in Python are explained in Table 1.1.

Table 1.1 Built-in exceptions in Python

S. No	Name of the Built-in Exception	Explanation
1.	SyntaxError	It is raised when there is an error in the syntax of the Python code.
2.	ValueError	It is raised when a built-in method or operation receives an argument that has the right data type but mismatched or inappropriate values.
3.	IOError	It is raised when the file specified in a program statement cannot be opened.

© NCERT not to be republished

Reprint 2026-27

4	KeyboardInterrupt	It is raised when the user accidentally hits the Delete or Esc key while executing a program due to which the normal flow of the program is interrupted.
5	ImportError	It is raised when the requested module definition is not found.
6	EOFError	It is raised when the end of file condition is reached without reading any data by input().
7	ZeroDivisionError	It is raised when the denominator in a division operation is zero.
8	IndexError	It is raised when the index or subscript in a sequence is out of range.
9	NameError	It is raised when a local or global variable name is not defined.
10	IndentationError	It is raised due to incorrect indentation in the program code.
11	TypeError	It is raised when an operator is supplied with a value of incorrect data type.
12	OverFlowError	It is raised when the result of a calculation exceeds the maximum limit for numeric data type.

Figure 1.4 shows the built-in exceptions viz, ZeroDivisionError, NameEError, and TypeError raised by the Python interpreter in different situations.

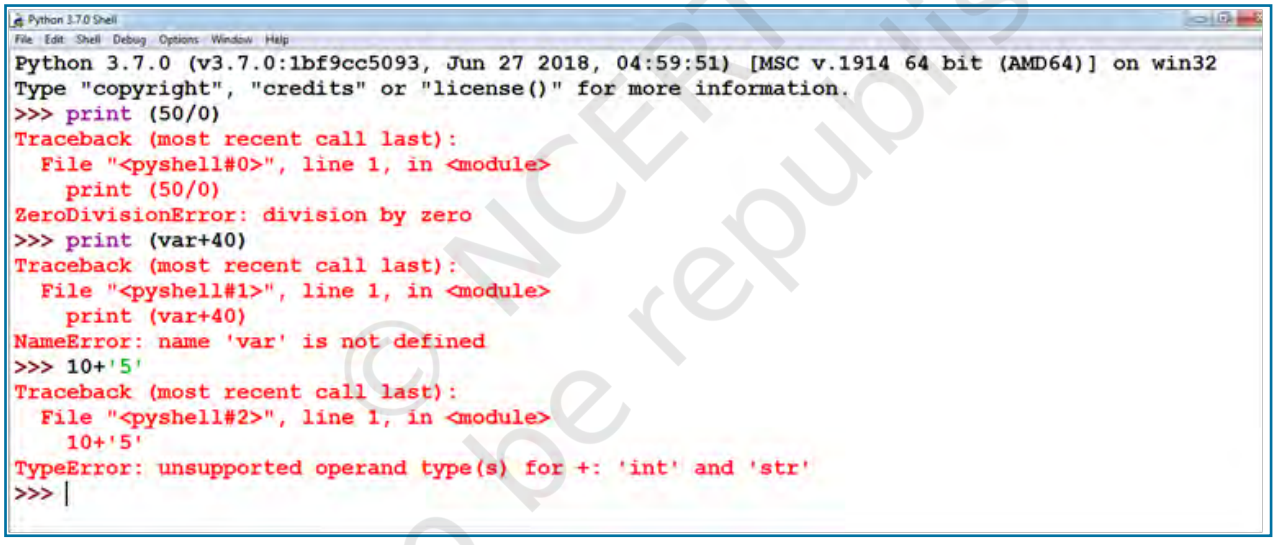


Figure 1.4: Example of built-in exceptions

A programmer can also create custom exceptions to suit one's requirements. These are called user-defined exceptions. We will learn how to handle exceptions in the next section.

1.5 Raising Exceptions

Each time an error is detected in a program, the Python interpreter raises (throws) an exception. Exception

© NCERT not to be republished

handlers are designed to execute when a specific exception is raised. Programmers can also forcefully raise exceptions in a program using the raise and assert statements. Once an exception is raised, no further statement in the current block of code is executed. So, raising an exception involves interrupting the normal flow execution of program and jumping to that part of the program (exception handler code) which is written to handle such exceptional situations.

1.5.1 The raise Statement

The raise statement can be used to throw an exception. The syntax of raise statement is:

```
raise exception-name[(optional argument)]
```

The argument is generally a string that is displayed when the exception is raised. For example, when an exception is raised as shown in Figure 1.5, the message "OOPS : An Exception has occurred" is displayed along with a brief description of the error.

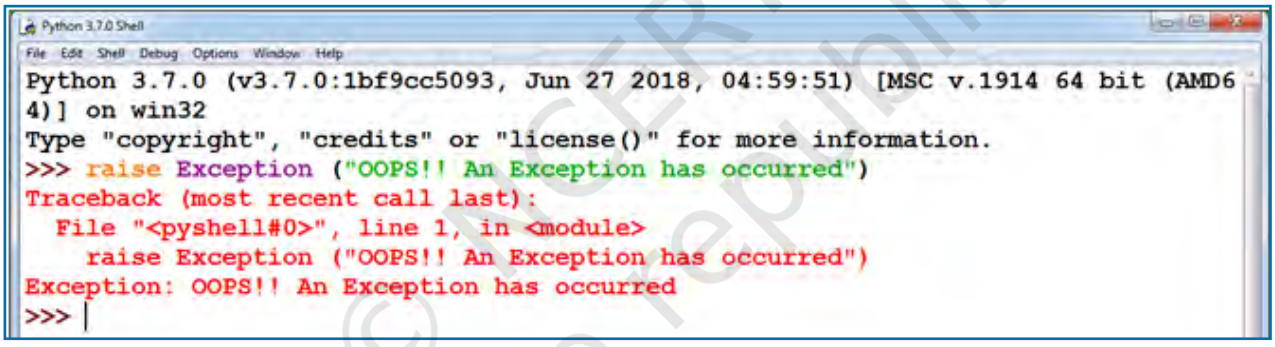


Figure 1.5: Use of the raise statement to throw an exception

The error detected may be a built-in exception or may be a user-defined one. Consider the example given in Figure 1.6 that uses the `raise` statement to raise a built-in exception called `IndexError`.

Note: In this case, the user has only raised the exception but has not displayed any error message explicitly.

In Figure 1.6, since the value of variable length is greater than the length of the list *numbers*, an `IndexError` exception will be raised. The statement following the raise statement will not be executed. So the message "NO EXECUTION" will not be displayed in this case.

© NCERT not to be republished

As we can see in Figure 1.6, in addition to the error message displayed, Python also displays a stack Traceback. This is a structured block of text that contains information about the sequence of function calls that have been made in the branch of execution of code in which the exception was raised. In Figure 1.6, the error has been encountered in the most recently called function that has been executed.

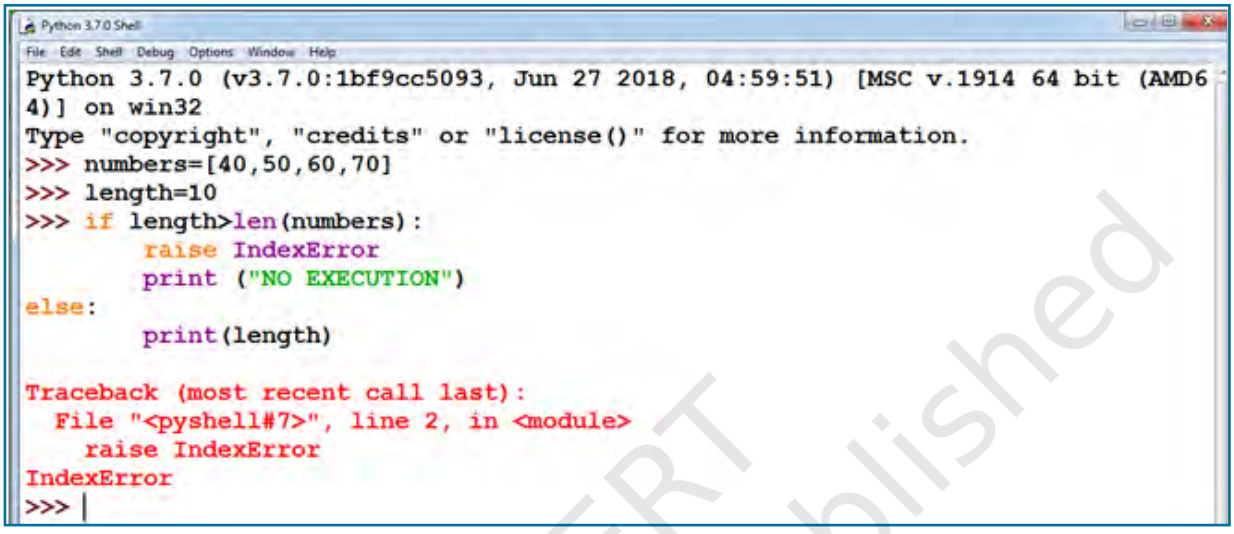


Figure 1.6: Use of raise statement with built-in exception

Note: We will learn about Stack in Chapter 3.

1.5.2 The assert Statement

An assert statement in Python is used to test an expression in the program code. If the result after testing comes false, then the exception is raised. This statement is generally used in the beginning of the function or after a function call to check for valid input. The syntax for assert statement is:

```
assert Expression[,arguments]
```

On encountering an assert statement, Python evaluates the expression given immediately after the assert keyword. If this expression is false, an AssertionError exception is raised which can be handled like any other exception. Consider the code given in Program 1-1.

Program 1-1 Use of assert statement

```
print("use of assert statement")
def negativecheck(number):
    assert(number>=0), "OOPS... Negative Number"
```

© NCERT not to be republished

```
        print(number*number)
    print (negativecheck(100))
    print (negativecheck(-350))
```

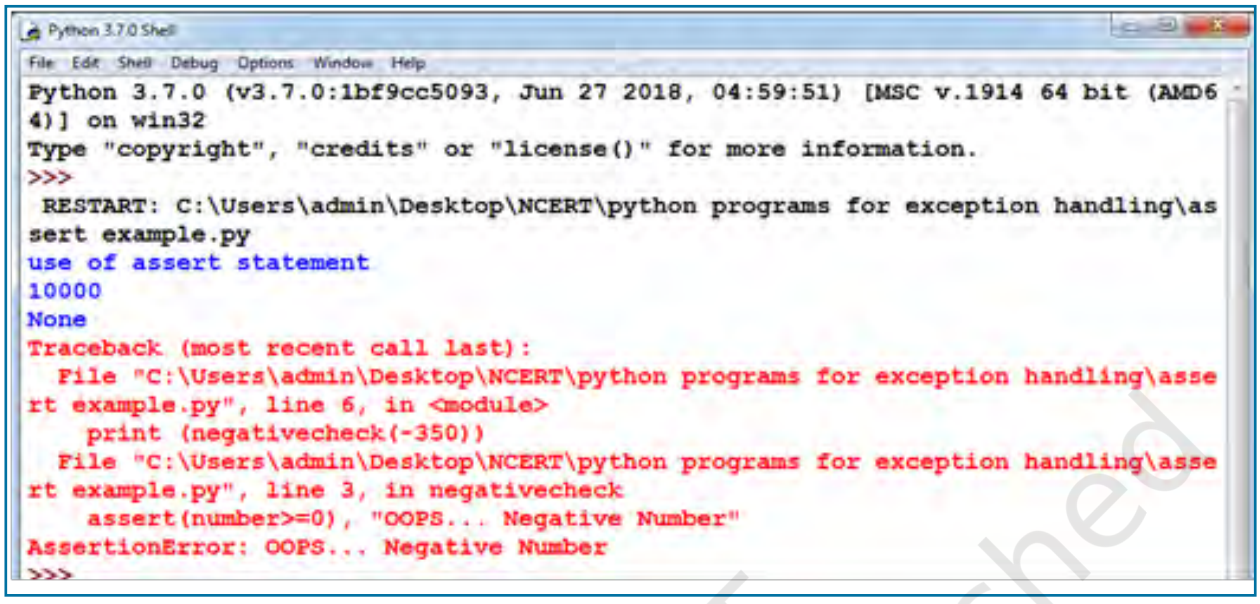


Figure 1.7: Output of Program 1-1.

In the code, the assert statement checks for the value of the variable number. In case the number gets a negative value, `AssertionError` will be thrown, and subsequent statements will not be executed. Hence, on passing a negative value (-350) as an argument, it results in `AssertionError` and displays the message "OOPS.... Negative Number". The output of the code is shown in Figure 1.7.

1.6 Handling Exceptions

Each and every exception has to be handled by the programmer to avoid the program from crashing abruptly. This is done by writing additional code in a program to give proper messages or instructions to the user on encountering an exception. This process is known as *exception handling*.

1.6.1 Need for Exception Handling

Exception handling is being used not only in Python programming but in most programming languages like C++, Java, Ruby, etc. It is a useful technique that helps in capturing runtime errors and handling them so as to avoid the program getting crashed. Following are some

© NCERT not to be republished

Reprint 2026-27

of the important points regarding exceptions and their handling:

- Python categorises exceptions into distinct types so that specific exception handlers (code to handle that particular exception) can be created for each type.
- Exception handlers separate the main logic of the program from the error detection and correction code. The segment of code where there is any possibility of error or exception, is placed inside one block. The code to be executed in case the exception has occurred, is placed inside another block. These statements for detection and reporting the exception do not affect the main logic of the program.
- The compiler or interpreter keeps track of the exact position where the error has occurred.
- Exception handling can be done for both user-defined and built-in exceptions.

1.6.2 Process of Handling Exception

When an error occurs, Python interpreter creates an object called the *exception object*. This object contains information about the error like its type, file name and position in the program where the error has occurred. The object is handed over to the runtime system so that it can find an appropriate code to handle this particular exception. This process of creating an exception object and handing it over to the runtime system is called *throwing* an exception. It is important to note that when an exception occurs while executing a particular program statement, the control jumps to an exception handler, abandoning execution of the remaining program statements.

A runtime system refers to the execution of the statements given in the program. It is a complex mechanism consisting of hardware and software that comes into action as soon as the program, written in any programming language, is put for execution.

The runtime system searches the entire program for a block of code, called the *exception handler* that can handle the raised exception. It first searches for the method in which the error has occurred and the exception has been raised. If not found, then it searches the method from which this method (in which exception was raised) was called. This hierarchical search in reverse order continues till the exception handler is found. This entire list of methods is known as *call stack*. When a suitable handler is found in the call stack, it is executed by the runtime process. This process of

© NCERT not to be republished

executing a suitable handler is known as *catching the exception*. If the runtime system is not able to find an appropriate exception after searching all the methods in the call stack, then the program execution stops.

The flowchart in Figure 1.8 describes the exception handling process.

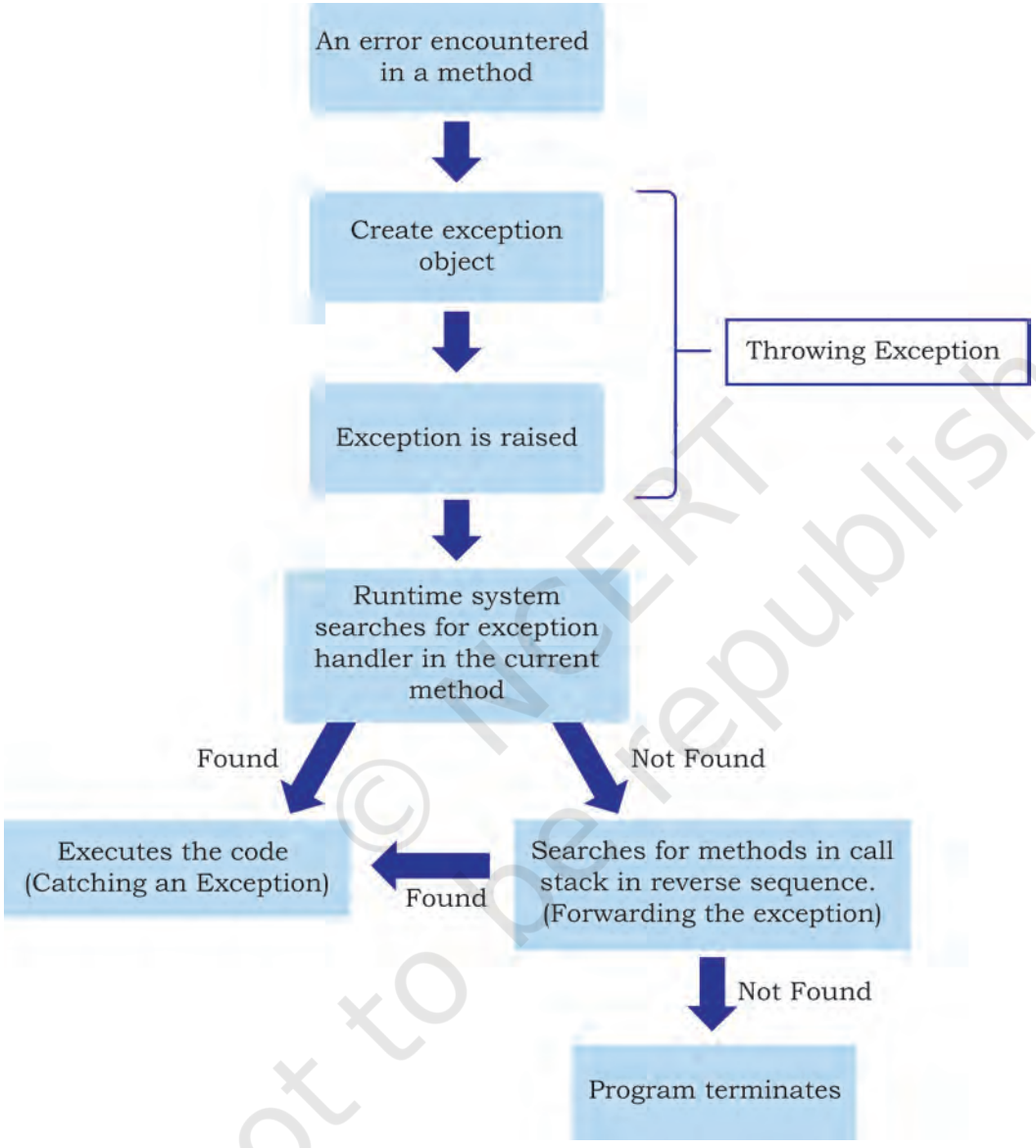


Figure 1.8: Steps of handling exception

1.6.3 Catching Exceptions

An exception is said to be caught when a code that is designed to handle a particular exception is executed. Exceptions, if any, are caught in the `try` block and

© NCERT not to be republished

handled in the except block. While writing or debugging a program, a user might doubt an exception to occur in a particular part of the code. Such suspicious lines of codes are put inside a try block. Every try block is followed by an except block. The appropriate code to handle each of the possible exceptions (in the code inside the try block) are written inside the except clause.

While executing the program, if an exception is encountered, further execution of the code inside the try block is stopped and the control is transferred to the except block. The syntax of try … except clause is as follows:

```
try:
   [ program statements where exceptions might occur]
except [exception-name]:
   [ code for exception handling if the exception-name error is
   encountered]
```

Consider the Program 1-2 given below:

Program 1-2 Using try..except block

```
print ("Practicing for try block")
try:
    numerator=50
    denom=int(input("Enter the denominator"))
    quotient=(numerator/denom)
    print(quotient)
    print ("Division performed successfully")
except ZeroDivisionError:
    print ("Denominator as ZERO.... not allowed")
print("OUTSIDE try..except block")
```

In Program 1-2, the ZeroDivisionError exception is handled. If the user enters any non-zero value as denominator, the quotient will be displayed along with the message "Division performed successfully", as shown in Figure 1.10. The except clause will be skipped in this case. So, the next statement after the try..except block is executed and the message "OUTSIDE try.. except block" is displayed.

However, if the user enters the value of denom as zero (0), then the execution of the try block will stop. The control will shift to the except block and the message "Denominator as Zero…. not allowed" will be displayed, as shown in Figure 1.11. Thereafter, the

© NCERT not to be republished

statement following the try..except block is executed and the message "`OUTSIDE try..except block`" is displayed in this case also.

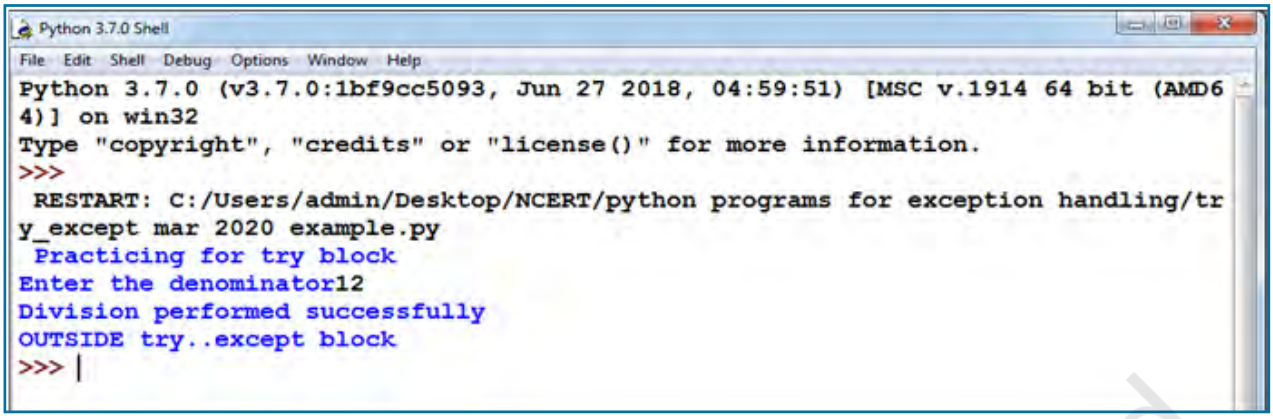


Figure 1.9: Output without an error

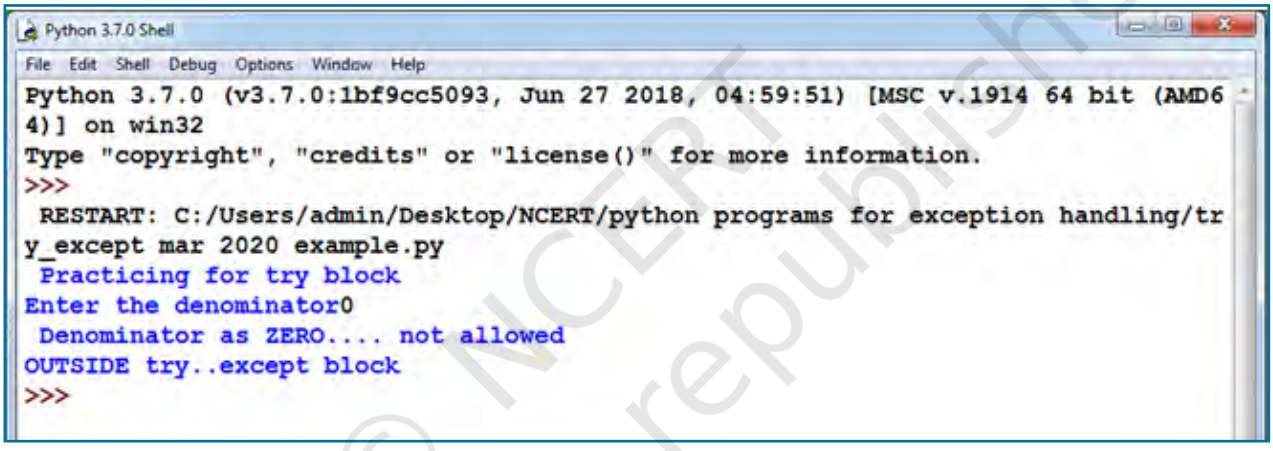


Figure 1.10: Output with exception raised

Sometimes, a single piece of code might be suspected to have more than one type of error. For handling such situations, we can have multiple except blocks for a single try block as shown in the Program 1-3.

Program 1-3 Use of multiple except clauses

```
print ("Handling multiple exceptions")
try:
    numerator=50
    denom=int(input("Enter the denominator: "))
    print (numerator/denom)
    print ("Division performed successfully")
except ZeroDivisionError:
    print ("Denominator as ZERO is not allowed")
except ValueError:
    print ("Only INTEGERS should be entered")
```

© NCERT not to be republished

In the code, two types of exceptions (ZeroDivisionError and `ValueError`) are handled using two `except` blocks for a single `try` block. When an exception is raised, a search for the matching `except` block is made till it is handled. If no match is found, then the program terminates.

However, if an exception is raised for which no handler is created by the programmer, then such an exception can be handled by adding an except clause without specifying any exception. This except clause should be added as the last clause of the try..except block. The Program 1-4 given below along with the output given in Figure 1.11 explains this.

Program 1-4 Use of except without specifying an exception

```
print ("Handling exceptions without naming them")
try:
    numerator=50
    denom=int(input("Enter the denominator"))
    quotient=(numerator/denom)
    print ("Division performed successfully")
except ValueError:
    print ("Only INTEGERS should be entered")
except:
    print(" OOPS.....SOME EXCEPTION RAISED")
```

If the above code is executed, and the denominator entered is 0 (zero) , the handler for `ZeroDivisionError` exception will be searched. Since it is not present, the last except clause (without any specified exception) will be executed , so the message " OOPS.....SOME EXCEPTION RAISED" will be displayed.

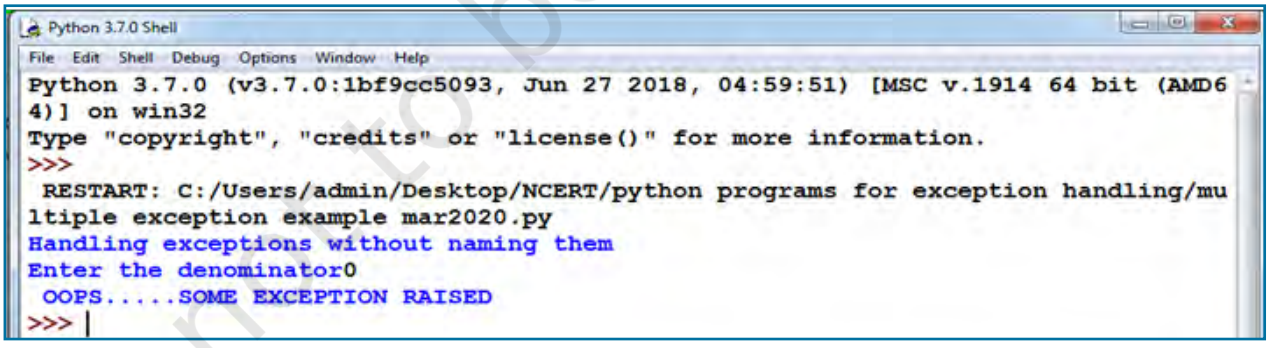


Figure 1.11: Output of Program 1-4

1.6.4 `try...except…else` clause

We can put an optional `else` clause along with the `try...except` clause. An `except` block will be executed

© NCERT not to be republished

only if some exception is raised in the `try` block. But if there is no error then none of the `except` blocks will be executed. In this case, the statements inside the `else` clause will be executed. Program 1-5 along with its output explains the use of else block with the try... except block.

Program 1-5 Use of else clause

```
print ("Handling exception using try...except...else")
try:
    numerator=50
    denom=int(input("Enter the denominator: "))
    quotient=(numerator/denom)
    print ("Division performed successfully")

except ZeroDivisionError:
    print ("Denominator as ZERO is not allowed")

except ValueError:
    print ("Only INTEGERS should be entered")

else:
   print ("The result of division operation is ", quotient)
```

Output:

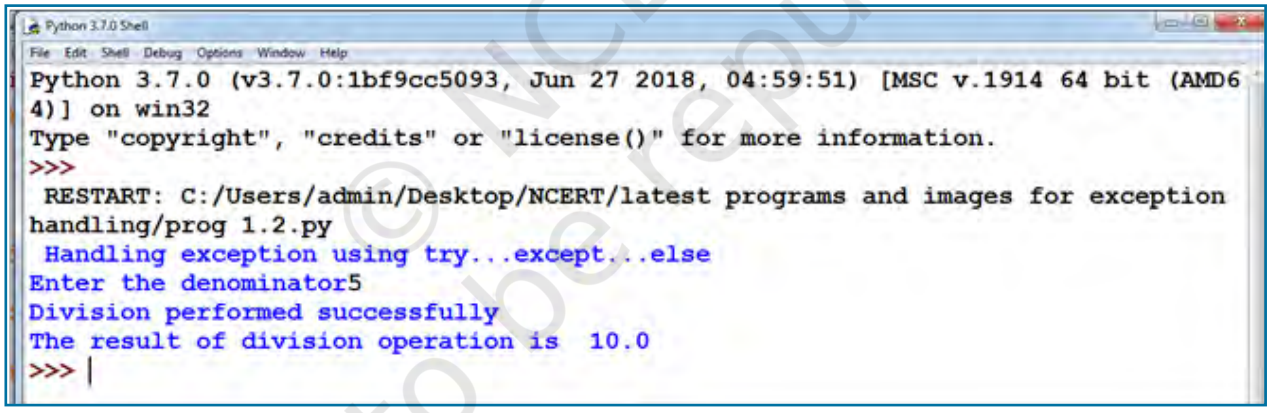


Figure 1.12: Output of Program 1-5.

1.7 Finally Clause

The try statement in Python can also have an optional finally clause. The statements inside the finally block are always executed regardless of whether an exception has occurred in the try block or not. It is a common practice to use `finally` clause while working with files to ensure that the file object is closed. If used, `finally` should always be placed at the end of `try` clause, after all `except` blocks and the `else` block.

© NCERT not to be republished

Reprint 2026-27

Program 1-6 Use of finally clause

```
print ("Handling exception using try...except...else...finally")
try:
  numerator=50
  denom=int(input("Enter the denominator: "))
  quotient=(numerator/denom)
  print ("Division performed successfully")
except ZeroDivisionError:
  print ("Denominator as ZERO is not allowed")
except ValueError:
  print ("Only INTEGERS should be entered")
else:
  print ("The result of division operation is ", quotient)
finally:
  print ("OVER AND OUT")
```

In the above program, the message "OVER AND OUT" will be displayed irrespective of whether an exception is raised or not.

1.7.1 Recovering and continuing with finally clause

If an error has been detected in the `try` block and the exception has been thrown, the appropriate except block will be executed to handle the error. But if the exception is not handled by any of the `except` clauses, then it is re-raised after the execution of the `finally` block. For example, Program 1.4 contains only the `except` block for `ZeroDivisionError`. If any other type of error occurs for which there is no handler code (except clause) defined, then also the `finally` clause will be executed first. Consider the code given in Program 1-7 to understand these concepts.

Program 1-7 Recovering through finally clause

```
print (" Practicing for try block")
try:
  numerator=50
  denom=int(input("Enter the denominator"))
  quotient=(numerator/denom)
  print ("Division performed successfully")
except ZeroDivisionError:
  print ("Denominator as ZERO is not allowed")
else:
  print ("The result of division operation is ", quotient)
finally:
  print ("OVER AND OUT")
```

© NCERT not to be republished

While executing the above code, if we enter a non-numeric data as input, the `finally` block will be executed. So, the message "OVER AND OUT" will be displayed. Thereafter the exception for which handler is not present will be re-raised. The output of Program 1-7 is shown in Figure 1.13.

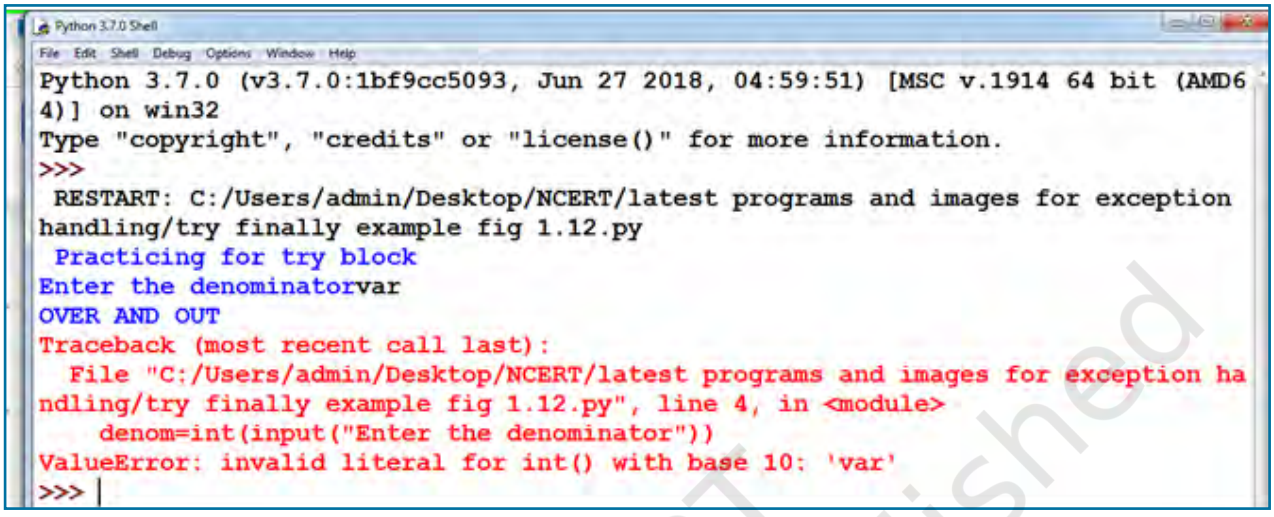


Figure 1.13: Output of Program 1-7

After execution of `finally` block, Python transfers the control to a previously entered try or to the next higher level default exception handler. In such a case, the statements following the `finally` block is executed. That is, unlike except, execution of the finally clause does not terminate the exception. Rather, the exception continues to be raised after execution of finally.

To summarise, we put a piece of code where there are possibilities of errors or exceptions to occur inside a try block. Inside each except clause we define handler codes to handle the matching exception raised in the try block. The optional else clause contains codes to be executed if no exception occurs. The optional finally block contains codes to be executed irrespective of whether an exception occurs or not.

SUMMARY

- Syntax errors or parsing errors are detected when we have not followed the rules of the particular programming language while writing a program.

© NCERT not to be republished

- When syntax error is encountered, Python displays the name of the error and a small description about the error.
- The execution of the program will start only after the syntax error is rectified.
- An exception is a Python object that represents an error.
- Syntax errors are also handled as exceptions.
- The exception needs to be handled by the programmer so that the program does not terminate abruptly.
- When an exception occurs during execution of a program and there is a built-in exception defined for that, the error message written in that exception is displayed. The programmer then has to take appropriate action and handle it.
- Some of the commonly occurring built-in exceptions are `SyntaxError`, `ValueError`, `IOError`, `KeyboardInterrupt`, `ImportError`, `EOFError`, `ZeroDivisionError`, `IndexError`, `NameError`, `IndentationError`, `TypeError`, and `OverFlowerror`.
- When an error is encountered in a program, Python interpreter raises or throws an exception.
- Exception Handlers are the codes that are designed to execute when a specific exception is raised.
- Raising an exception involves interrupting the normal flow of the program execution and jumping to the exception handler.
- Raise and assert statements are used to raise exceptions.
- The process of exception handling involves writing additional code to give proper messages or instructions to the user. This prevents the program from crashing abruptly. The additional code is known as an exception handler.
- An exception is said to be caught when a code that is designed to handle a particular exception is executed.
- An exception is caught in the try block and handles in except block.

© NCERT not to be republished

- The statements inside the finally block are always executed regardless of whether an exception occurred in the try block or not.

Exercise

1. "Every syntax error is an exception but every exception cannot be a syntax error." Justify the statement.
2. When are the following built-in exceptions raised? Give examples to support your answers.
   a) ImportError
   b) IOError
   c) NameError
   d) ZeroDivisionError
3. What is the use of a raise statement? Write a code to accept two numbers and display the quotient. Appropriate exception should be raised if the user enters the second number (denominator) as zero (0).
4. Use assert statement in Question No. 3 to test the division expression in the program.
5. Define the following:
   a) Exception Handling
   b) Throwing an exception
   c) Catching an exception
6. Explain catching exceptions using try and except block.
7. Consider the code given below and fill in the blanks.

```
print (" Learning Exceptions...")
try:
    num1= int(input ("Enter the first number"))
    num2=int(input("Enter the second number"))
    quotient=(num1/num2)
    print ("Both the numbers entered were correct")
except _____:              # to enter only integers
    print (" Please enter only numbers")
except _____:      # Denominator should not be zero
  print(" Number 2 should not be zero")
else:
  print(" Great .. you are a good programmer")
_____:               # to be executed at the end
  print(" JOB OVER... GO GET SOME REST")
```

© NCERT not to be republished

Reprint 2026-27

8. You have learnt how to use math module in Class XI. Write a code where you use the wrong number of arguments for a method (say sqrt() or pow()). Use the exception handling process to catch the ValueError exception.
9. What is the use of finally clause? Use finally clause in the problem given in Question No. 7.

© NCERT not to be republished

Chapter 2

File Handling in Python

12130CH02

There are many ways of trying to understand programs. People often rely too much on one way, which is called "debugging" and consists of running a partly-understood program to see if it does what you expected. Another way, which ML advocates, is to install some means of understanding in the very programs themselves.

— Robin Milner

In this Chapter

- *Introduction to Files*
- *Types of Files*
- *Opening and Closing a Text File*
- *Writing to a Text File*
- *Reading from a Text File*
- *Setting Offsets in a File*
- *Creating and Traversing a Text File*
- *The Pickle Module*

2.1 Introduction to Files

We have so far created programs in Python that accept the input, manipulate it and display the output. But that output is available only during execution of the program and input is to be entered through the keyboard. This is because the variables used in a program have a lifetime that lasts till the time the program is under execution. What if we want to store the data that were input as well as the generated output permanently so that we can reuse it later? Usually, organisations would want to permanently store information about employees, inventory, sales, etc. to avoid repetitive tasks of entering the same data. Hence, data are stored permanently on secondary storage devices for reusability. We store Python programs written in script mode with a .py extension. Each program is stored on the secondary device as a file. Likewise, the data entered, and the output can be stored permanently into a file.

© NCERT not to be republished

Text files contain only the ASCII equivalent of the contents of the file whereas a .docx file contains many additional information like the author's name, page settings, font type and size, date of creation and modification, etc.

So, what is a file? A file is a named location on a secondary storage media where data are permanently stored for later access.

2.2. Types of Files

Computers store every file as a collection of 0s and 1s i.e., in binary form. Therefore, every file is basically just a series of bytes stored one after the other. There are mainly two types of data files — text file and binary file. A text file consists of human readable characters, which can be opened by any text editor. On the other hand, binary files are made up of non-human readable characters and symbols, which require specific programs to access its contents.

2.2.1 Text file

A text file can be understood as a sequence of characters consisting of alphabets, numbers and other special symbols. Files with extensions like .txt, .py, .csv, etc. are some examples of text files. When we open a text file using a text editor (e.g., Notepad), we see several lines of text. However, the file contents are not stored in such a way internally. Rather, they are stored in sequence of bytes consisting of 0s and 1s. In ASCII, UNICODE or any other encoding scheme, the value of each character of the text file is stored as bytes. So, while opening a text file, the text editor translates each ASCII value and shows us the equivalent character that is readable by the human being. For example, the ASCII value 65 (binary equivalent 1000001) will be displayed by a text editor as the letter 'A' since the number 65 in ASCII character set represents 'A'.

Each line of a text file is terminated by a special character, called the End of Line (EOL). For example, the default EOL character in Python is the newline (\n). However, other characters can be used to indicate EOL. When a text editor or a program interpreter encounters the ASCII equivalent of the EOL character, it displays the remaining file contents starting from a new line. Contents in a text file are usually separated by whitespace, but comma (,) and tab (\t) are also commonly used to separate values in a text file.

Activity 2.1

Create a text file using notepad and write your name and save it. Now, create a .docx file using Microsoft Word and write your name and save it as well. Check and compare the file size of both the files. You will find that the size of .txt file is in bytes whereas that of .docx is in KBs.

© NCERT not to be republished

Reprint 2026-27

2.2.2 Binary Files

Binary files are also stored in terms of bytes (0s and 1s), but unlike text files, these bytes do not represent the ASCII values of characters. Rather, they represent the actual content such as image, audio, video, compressed versions of other files, executable files, etc. These files are not human readable. Thus, trying to open a binary file using a text editor will show some garbage values. We need specific software to read or write the contents of a binary file.

Binary files are stored in a computer in a sequence of bytes. Even a single bit change can corrupt the file and make it unreadable to the supporting application. Also, it is difficult to remove any error which may occur in the binary file as the stored contents are not human readable. We can read and write both text and binary files through Python programs.

2.3 Opening and Closing a Text File

In real world applications, computer programs deal with data coming from different sources like databases, CSV files, HTML, XML, JSON, etc. We broadly access files either to write or read data from it. But operations on files include creating and opening a file, writing data in a file, traversing a file, reading data from a file and so on. Python has the io module that contains different functions for handling files.

2.3.1 Opening a file

To open a file in Python, we use the open() function. The syntax of open() is as follows:

```
file_object= open(file_name, access_mode)
```

This function returns a file object called file handle which is stored in the variable `file_object`. We can use this variable to transfer data to and from the file (read and write) by calling the functions defined in the Python's `io module`. If the file does not exist, the above statement creates a new empty file and assigns it the name we specify in the statement.

The `file_object` has certain attributes that tells us basic information about the file, such as:

- `<file.closed>` returns true if the file is closed and false otherwise.

The `file_object` establishes a link between the program and the data file stored in the permanent storage.

© NCERT not to be republished

- <file.mode> returns the access mode in which the file was opened.
- <file.name> returns the name of the file.

Activity 2.2

Some of the other file access modes are <rb+>, <wb>, <w+>, <ab>, <ab+>. Find out for what purpose each of these are used. Also, find the file offset positions in each case.

The file_name should be the name of the file that has to be opened. If the file is not in the current working directory, then we need to specify the complete path of the file along with its name.

The access_mode is an optional argument that represents the mode in which the file has to be accessed by the program. It is also referred to as processing mode. Here mode means the operation for which the file has to be opened like <r> for reading, <w> for writing, <+> for both reading and writing, <a> for appending at the end of an existing file. The default is the read mode. In addition, we can specify whether the file will be handled as binary (<b>) or text mode. By default, files are opened in text mode that means strings can be read or written. Files containing non-textual data are opened in binary mode that means read/write are performed in terms of bytes. Table 2.1 lists various file access modes that can be used with the open() method. The file offset position in the table refers to the position of the file object when the file is opened in a particular mode.

Table 2.1 File Open Modes

File Mode	Description	File Offset position
<r>	Opens the file in read-only mode.	Beginning of the file
<rb>	Opens the file in binary and read-only mode.	Beginning of the file
<r+> or <+r>	Opens the file in both read and write mode.	Beginning of the file
<w>	Opens the file in write mode. If the file already exists, all the contents will be overwritten. If the file doesn't exist, then a new file will be created.	Beginning of the file
<wb+> or <+wb>	Opens the file in read,write and binary mode. If the file already exists, the contents will be overwritten. If the file doesn't exist, then a new file will be created.	Beginning of the file
<a>	Opens the file in append mode. If the file doesn't exist, then a new file will be created.	End of the file
<a+> or <+a>	Opens the file in append and read mode. If the file doesn't exist, then it will create a new file.	End of the file

Consider the following example.

```
myObject=open("myfile.txt", "a+")
```

© NCERT not to be republished

Reprint 2026-27

In the above statement, the file *myfile.txt* is opened in append and read modes. The file object will be at the end of the file. That means we can write data at the end of the file and at the same time we can also read data from the file using the file object named *myObject*.

2.3.2 Closing a file

Once we are done with the read/write operations on a file, it is a good practice to close the file. Python provides a `close()` method to do so. While closing a file, the system frees the memory allocated to it. The syntax of `close()` is:

```
file_object.close()
```

Here, file_object is the object that was returned while opening the file.

Python makes sure that any unwritten or unsaved data is flushed off (written) to the file before it is closed. Hence, it is always advised to close the file once our work is done. Also, if the file object is re-assigned to some other file, the previous file is automatically closed.

2.3.3 Opening a file using with clause

In Python, we can also open a file using with clause. The syntax of with clause is:

```
with open (file_name, access_mode) as file_
object:
```

The advantage of using with clause is that any file that is opened using this clause is closed automatically, once the control comes outside the with clause. In case the user forgets to close the file explicitly or if an exception occurs, the file is closed automatically. Also, it provides a simpler syntax.

```
with open("myfile.txt","r+") as myObject:
    content = myObject.read()
```

Here, we don't have to close the file explicitly using close() statement. Python will automatically close the file.

2.4 Writing to a Text File

For writing to a file, we first need to open it in write or append mode. If we open an existing file in write mode, the previous data will be erased, and the file object will be positioned at the beginning of the file. On the other

© NCERT not to be republished

Reprint 2026-27

Think and Reflect

For a newly created file, is there any difference between write() and append() methods?

hand, in append mode, new data will be added at the end of the previous data as the file object is at the end of the file. After opening the file, we can use the following methods to write data in the file.

- write() - for writing a single string
- writelines() - for writing a sequence of strings

2.4.1 The write() method

write() method takes a string as an argument and writes it to the text file. It returns the number of characters being written on single execution of the write() method. Also, we need to add a newline character (\n) at the end of every sentence to mark the end of line.

Consider the following piece of code:

```
>>> myobject=open("myfile.txt",'w')
>>> myobject.write("Hey I have started
    #using files in Python\n")
41
>>> myobject.close()
```

On execution, write() returns the number of characters written on to the file. Hence, 41, which is the length of the string passed as an argument, is displayed.

Note: '\n' is treated as a single character

We can also use the flush() method to clear the buffer and write contents in buffer to the file. This is how programmers can forcefully write to the file as and when required.

If numeric data are to be written to a text file, the data need to be converted into string before writing to the file. For example:

```
>>>myobject=open("myfile.txt",'w')
>>> marks=58
#number 58 is converted to a string using
#str()
>>> myobject.write(str(marks))
2
>>>myobject.close()
```

The `write()` actually writes data onto a buffer. When the close() method is executed, the contents from this buffer are moved to the file located on the permanent storage.

2.4.2 The writelines() method

This method is used to write multiple strings to a file. We need to pass an iterable object like lists, tuple, etc. containing strings to the `writelines()` method. Unlike

© NCERT not to be republished

write(), the writelines() method does not return the number of characters written in the file. The following code explains the use of writelines().

```
>>> myobject=open("myfile.txt",'w')
>>> lines = ["Hello everyone\n", "Writing
    #multiline strings\n", "This is the
    #third line"]
>>> myobject.writelines(lines)
>>>myobject.close()
```

On opening myfile.txt, using notepad, its content will appear as shown in Figure 2.1.

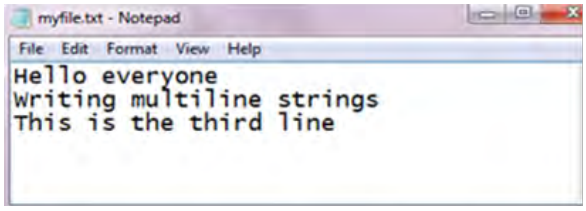


Figure 2.1: Contents of myfile.txt

2.5 Reading from a Text File

We can write a program to read the contents of a file. Before reading a file, we must make sure that the file is opened in "r", "r+", "w+" or "a+" mode. There are three ways to read the contents of a file:

2.5.1 The read() method

This method is used to read a specified number of bytes of data from a data file. The syntax of read() method is:

```
file_object.read(n)
```

Consider the following set of statements to understand the usage of read() method:

```
>>>myobject=open("myfile.txt",'r')
>>> myobject.read(10)
'Hello ever'
>>> myobject.close()
```

If no argument or a negative number is specified in read(), the entire file content is read. For example,

```
>>> myobject=open("myfile.txt",'r')
>>> print(myobject.read())
Hello everyone
Writing multiline strings
This is the third line
>>> myobject.close()
```

Activity 2.3

Run the above code by replacing writelines() with write() and see what happens.

Think and Reflect

Can we pass a tuple of numbers as an argument to writelines()? Will it be written to the file or an error will be generated?

© NCERT not to be republished

2.5.2 The readline([n]) method

This method reads one complete line from a file where each line terminates with a newline (\n) character. It can also be used to read a specified number (n) of bytes of data from a file but maximum up to the newline character (\n). In the following example, the second statement reads the first ten characters of the first line of the text file and displays them on the screen.

```
>>> myobject=open("myfile.txt",'r')
>>> myobject.readline(10)
'Hello ever'
>>> myobject.close()
```

If no argument or a negative number is specified, it reads a complete line and returns string.

```
>>>myobject=open("myfile.txt",'r')
>>> print (myobject.readline())
'Hello everyone\n'
```

Activity 2.4

Create a file having multiline data and use readline() with an iterator to read the contents of the file line by line

To read the entire file line by line using the readline(), we can use a loop. This process is known as looping/ iterating over a file object. It returns an empty string when EOF is reached.

2.5.3 The readlines() method

The method reads all the lines and returns the lines along with newline as a list of strings. The following example uses readlines() to read data from the text file *myfile.txt*.

```
>>> myobject=open("myfile.txt", 'r')
>>> print(myobject.readlines())
['Hello everyone\n', 'Writing multiline
strings\n', 'This is the third line']
>>> myobject.close()
```

As shown in the above output, when we read a file using readlines() function, lines in the file become members of a list, where each list element ends with a newline character ('\n').

In case we want to display each word of a line separately as an element of a list, then we can use split() function. The following code demonstrates the use of split() function.

```
>>> myobject=open("myfile.txt",'r')
>>> d=myobject.readlines()
```

© NCERT not to be republished

```
>>> for line in d:
        words=line.split()
        print(words)

['Hello', 'everyone']
['Writing', 'multiline', 'strings']
['This', 'is', 'the', 'third', 'line']
```

In the output, each string is returned as elements of a list. However, if *splitlines()* is used instead of split(), then each line is returned as element of a list, as shown in the output below:

```
>>> for line in d:
        words=line.splitlines()
        print(words)

['Hello everyone']
['Writing multiline strings']
['This is the third line']
```

Let us now write a program that accepts a string from the user and writes it to a text file. Thereafter, the same program reads the text file and displays it on the screen.

Program 2-1 Writing and reading to a text file

```
fobject=open("testfile.txt","w")    # creating a data file
sentence=input("Enter the contents to be written in the file: ")
fobject.write(sentence)             # Writing data to the file
fobject.close()                     # Closing a file

print("Now reading the contents of the file: ")
fobject=open("testfile.txt","r")
#looping over the file object to read the file
for str in fobject:
        print(str)
fobject.close()
```

In Program 2.1, the file named *testfile.txt* is opened in write mode and the file handle named *fobject* is returned. The string is accepted from the user and written in the file using write(). Then the file is closed and again opened in read mode. Data is read from the file and displayed till the end of file is reached.

© NCERT not to be republished

Output of Program 2-1:

```
>>>
 RESTART: Path_to_file\Program2-1.py
Enter the contents to be written in the file:
roll_numbers = [1, 2, 3, 4, 5, 6]
Now reading the contents of the file:
roll_numbers = [1, 2, 3, 4, 5, 6]
>>>
```

2.6 Setting Offsets in a File

The functions that we have learnt till now are used to access the data sequentially from a file. But if we want to access data in a random fashion, then Python gives us seek() and tell() functions to do so.

2.6.1 The tell() method

This function returns an integer that specifies the current position of the file object in the file. The position so specified is the byte position from the beginning of the file till the current position of the file object. The syntax of using tell() is:

```
file_object.tell()
```

2.6.2 The seek() method

This method is used to position the file object at a particular position in a file. The syntax of seek() is:

```
file_object.seek(offset [, reference_point])
```

In the above syntax, offset is the number of bytes by which the file object is to be moved. `reference_point` indicates the starting position of the file object. That is, with reference to which position, the offset has to be counted. It can have any of the following values:

0 - beginning of the file
1 - current position of the file
2 - end of file

By default, the value of `reference_point` is 0, i.e. the offset is counted from the beginning of the file. For example, the statement `fileObject.seek(5,0)` will position the file object at 5th byte position from the beginning of the file. The code in Program 2-2 below demonstrates the usage of seek() and tell().

Think and Reflect

Does the seek() function work in the same manner for text and binary files?

© NCERT not to be republished

Reprint 2026-27

Program 2-2 Application of seek() and tell()

```
print("Learning to move the file object")
fileobject=open("testfile.txt","r+")
str=fileobject.read()
print(str)
print("Initially, the position of the file object is: ",fileobject.
tell())
fileobject.seek(0)
print("Now the file object is at the beginning of the file:
",fileobject.tell())
fileobject.seek(10)
print("We are moving to 10th byte position from the beginning of
file")
print("The position of the file object is at", fileobject.tell())
str=fileobject.read()
print(str)
```

Output of Program 2-2:

```
>>>
 RESTART: Path_to_file\Program2-2.py
Learning to move the file object
roll_numbers = [1, 2, 3, 4, 5, 6]
Initially, the position of the file object is: 33
Now the file object is at the beginning of the file: 0
We are moving to 10th byte position from the beginning of file

The position of the file object is at 10
rs = [1, 2, 3, 4, 5, 6]
>>>
```

2.7 Creating and Traversing a Text File

Having learnt various methods that help us to open and close a file, read and write data in a text file, find the position of the file object and move the file object at a desired location, let us now perform some basic operations on a text file. To perform these operations, let us assume that we will be working with practice.txt.

2.7.1 Creating a file and writing data

To create a text file, we use the `open()` method and provide the filename and the mode. If the file already exists with the same name, the `open()` function will behave differently depending on the mode (write or append) used. If it is in write mode (w), then all the existing contents of file will be lost, and an empty file will be created with the same name. But, if the file is

© NCERT not to be republished

Reprint 2026-27

created in append mode (a), then the new data will be written after the existing data. In both cases, if the file does not exist, then a new empty file will be created. In Program 2-3, a file, *practice.txt* is opened in write (w) mode and three sentences are stored in it as shown in the output screen that follows it

Program 2-3 To create a text file and write data in it

```
# program to create a text file and add data
fileobject=open("practice.txt","w+")
while True:
    data= input("Enter data to save in the text file: ")
    fileobject.write(data)
    ans=input("Do you wish to enter more data?(y/n): ")
    if ans=='n': break
fileobject.close()
```

Output of Program 2-3:

```
>>>
 RESTART: Path_to_file\Program2-3.py
Enter data to save in the text file: I am interested to learn about
Computer Science
Do you wish to enter more data?(y/n): y
Enter data to save in the text file: Python is easy to learn
Do you wish to enter more data?(y/n): n
>>>
```

2.7.2 Traversing a file and displaying data

To read and display data that is stored in a text file, we will refer to the previous example where we have created the file *practice.txt.* The file will be opened in read mode and reading will begin from the beginning of the file.

Program 2-4 To display data from a text file

```
fileobject=open("practice.txt","r")
str = fileobject.readline()
while str:
    print(str)
    str=fileobject.readline()
fileobject.close()
```

In Program 2-4, the readline() is used in the while loop to read the data line by line from the text file. The lines are displayed using the print(). As the end of file is reached, the readline() will return an empty string. Finally, the file is closed using the close().

© NCERT not to be republished

Reprint 2026-27

Output of Program 2-4:

```
>>>
I am interested to learn about Computer SciencePython is easy to learn
```

Till now, we have been creating separate programs for writing data to a file and for reading the file. Now let us create one single program to read and write data using a single file object. Since both the operations have to be performed using a single file object, the file will be opened in w+ mode.

Program 2-5 To perform reading and writing operation in a text file

```
fileobject=open("report.txt", "w+")
print ("WRITING DATA IN THE FILE")
print() # to display a blank line
while True:
    line= input("Enter a sentence ")
    fileobject.write(line)
    fileobject.write('\n')
    choice=input("Do you wish to enter more data? (y/n): ")
    if choice in ('n','N'): break
print("The byte position of file object is ",fileobject.tell())
fileobject.seek(0)  #places file object at beginning of file
print()
print("READING DATA FROM THE FILE")
str=fileobject.read()
print(str)
fileobject.close()
```

In Program 2-5, the file will be read till the time end of file is not reached and the output as shown in below is displayed.

Output of Program 2-5:

```
>>>
 RESTART: Path_to_file\Program2-5.py
WRITING DATA IN THE FILE

Enter a sentence I am a student of class XII
Do you wish to enter more data? (y/n): y
Enter a sentence my school contact number is 4390xxx8
Do you wish to enter more data? (y/n): n
The byte position of file object is  67

READING DATA FROM THE FILE
I am a student of class XII
my school contact number is 4390xxx8
>>>
```

© NCERT not to be republished

2.8 THE PICKLE MODULE

We know that Python considers everything as an object. So, all data types including list, tuple, dictionary, etc. are also considered as objects. During execution of a program, we may require to store current state of variables so that we can retrieve them later to its present state. Suppose you are playing a video game, and after some time, you want to close it. So, the program should be able to store the current state of the game, including current level/stage, your score, etc. as a Python object. Likewise, you may like to store a Python dictionary as an object, to be able to retrieve later. To save any object structure along with data, Python provides a module called Pickle. The module Pickle is used for serializing and de-serializing any Python object structure. Pickling is a method of preserving food items by placing them in some solution, which increases the shelf life. In other words, it is a method to store food items for later consumption.

Serialization is the process of transforming data or an object in memory (RAM) to a stream of bytes called byte streams. These byte streams in a binary file can then be stored in a disk or in a database or sent through a network. Serialization process is also called pickling.

De-serialization or unpickling is the inverse of pickling process where a byte stream is converted back to Python object.

The pickle module deals with binary files. Here, data are not written but dumped and similarly, data are not read but loaded. The Pickle Module must be imported to load and dump data. The pickle module provides two methods - dump() and load() to work with binary files for pickling and unpickling, respectively.

2.8.1 The dump() method

This method is used to convert (pickling) Python objects for writing data in a binary file. The file in which data are to be dumped, needs to be opened in binary write mode (wb).

Syntax of `dump()` is as follows:

```
dump(data_object, file_object)
```

where `data_object` is the object that has to be dumped to the file with the file handle named `file_`

© NCERT not to be republished

object. For example, Program 2-6 writes the record of a student (roll_no, name, gender and marks) in the binary file named mybinary.dat using the dump(). We need to close the file after pickling.

Program 2-6 Pickling data in Python

```
import pickle
listvalues=[1,"Geetika",'F', 26]
fileobject=open("mybinary.dat", "wb")
pickle.dump(listvalues,fileobject)
fileobject.close()
```

2.8.2 The load() method

This method is used to load (unpickling) data from a binary file. The file to be loaded is opened in binary read (rb) mode. Syntax of load() is as follows:

Store_object = load(file_object)

Here, the pickled Python object is loaded from the file having a file handle named file_object and is stored in a new file handle called store_object. The program 2-7 demonstrates how to read data from the file *mybinary.dat* using the load().

Program 2-7 Unpickling data in Python

```
import pickle
print("The data that were stored in file are: ")
fileobject=open("mybinary.dat","rb")
objectvar=pickle.load(fileobject)
fileobject.close()
print(objectvar)
```

Output of Program 2-7:

```
>>>
 RESTART: Path_to_file\Program2-7.py
The data that were stored in file are:
[1, 'Geetika', 'F', 26]
>>>
```

2.8.3 File handling using pickle module

As we read and write data in a text file, similarly we will be adding and displaying data for a binary file. Program 2-8 accepts a record of an employee from the user and appends it in the binary file *tv*. Thereafter, the records are read from the binary file and displayed on the screen using the same object. The user may enter

© NCERT not to be republished

Reprint 2026-27

as many records as they wish to. The program also displays the size of binary files before starting with the reading process.

Program 2-8 To perform basic operations on a binary file using pickle module

```
# Program to write and read employee records in a binary file
import pickle
print("WORKING WITH BINARY FILES")
bfile=open("empfile.dat","ab")
recno=1
print ("Enter Records of Employees")
print()
#taking data from user and dumping in the file as list object
while True:
    print("RECORD No.", recno)
    eno=int(input("\tEmployee number : "))
    ename=input("\tEmployee Name : ")
    ebasic=int(input("\tBasic Salary : "))
    allow=int(input("\tAllowances : "))
    totsal=ebasic+allow
    print("\tTOTAL SALARY : ", totsal)
    edata=[eno,ename,ebasic,allow,totsal]
    pickle.dump(edata,bfile)
    ans=input("Do you wish to enter more records (y/n)? ")
    recno=recno+1
    if ans.lower()=='n':
        print("Record entry OVER ")
        print()
        break
# retrieving the size of file
print("Size of binary file (in bytes):",bfile.tell())
bfile.close()
# Reading the employee records from the file using load() module
print("Now reading the employee records from the file")
print()
readrec=1
try:
  with open("empfile.dat","rb") as bfile:
  while True:
       edata=pickle.load(bfile)
       print("Record Number : ",readrec)
       print(edata)
       readrec=readrec+1
except EOFError:
    pass
bfile.close()
```

© NCERT not to be republished

Output of Program 2-8:

```
>>>
 RESTART: Path_to_file\Program2-8.py
WORKING WITH BINARY FILES
Enter Records of Employees

RECORD No. 1
        Employee number : 11
        Employee Name : D N Ravi
        Basic Salary : 32600
        Allowances : 4400
        TOTAL SALARY :  37000
Do you wish to enter more records (y/n)? y
RECORD No. 2
        Employee number : 12
        Employee Name : Farida Ahmed
        Basic Salary : 38250
        Allowances : 5300
        TOTAL SALARY :  43550
Do you wish to enter more records (y/n)? n
Record entry OVER

Size of binary file (in bytes): 216
Now reading the employee records from the file

Record Number :  1
[11, 'D N Ravi', 32600, 4400, 37000]
Record Number :  2
[12, 'Farida Ahmed', 38250, 5300, 43550]
>>>
```

As each employee record is stored as a list in the file `empfile.dat`, hence while reading the file, a list is displayed showing record of each employee. Notice that in Program 2-8, we have also used try.. except block to handle the end-of-file exception.

SUMMARY

- A file is a named location on a secondary storage media where data are permanently stored for later access.
- A text file contains only textual information consisting of alphabets, numbers and other

© NCERT not to be republished

Reprint 2026-27

special symbols. Such files are stored with extensions like *.txt, .py, .c, .csv, .html,* etc. Each byte of a text file represents a character.

- Each line of a text file is stored as a sequence of ASCII equivalent of the characters and is terminated by a special character, called the End of Line (EOL).
- Binary file consists of data stored as a stream of bytes.
- open() method is used to open a file in Python and it returns a file object called file handle. The file handle is used to transfer data to and from the file by calling the functions defined in the Python's io module.
- close() method is used to close the file. While closing a file, the system frees up all the resources like processor and memory allocated to it.
- `write()` method takes a string as an argument and writes it to the text file.
- `writelines()` method is used to write multiple strings to a file. We need to pass an iterable object like lists, tuple etc. containing strings to writelines() method.
- `read([n])` method is used to read a specified number of bytes (n) of data from a data file.
- `readline([n])` method reads one complete line from a file where lines are ending with a newline (\n). It can also be used to read a specified number (n) of bytes of data from a file but maximum up to the newline character (\n).
- `readlines()` method reads all the lines and returns the lines along with newline character, as a list of strings.
- `tell()` method returns an integer that specifies the current position of the file object. The position so specified is the byte position from the beginning of the file till the current position of the file object.
- `seek()` method is used to position the file object at a particular position in a file.

© NCERT not to be republished

- Pickling is the process by which a Python object is converted to a byte stream.
- `dump()` method is used to write the objects in a binary file.
- `load()` method is used to read data from a binary file.

EXERCISE

1. Differentiate between:
   a) text file and binary file
   b) readline() and readlines()
   c) write() and writelines()

2. Write the use and syntax for the following methods:
   a) open()
   b) read()
   c) seek()
   d) dump()

3. Write the file mode that will be used for opening the following files. Also, write the Python statements to open the following files:
   a) a text file "example.txt" in both read and write mode
   b) a binary file "bfile.dat" in write mode
   c) a text file "try.txt" in append and read mode
   d) a binary file "btry.dat" in read only mode.

4. Why is it advised to close a file after we are done with the read and write operations? What will happen if we do not close it? Will some error message be flashed?

5. What is the difference between the following set of statements (a) and (b):
   a) P = open("practice.txt","r")
      P.read(10)
   b) with open("practice.txt", "r") as P:
         x = P.read()

6. Write a command(s) to write the following lines to the text file named *hello.txt*. Assume that the file is opened in append mode.
   " Welcome my class"
   "It is a fun place"
   "You will learn and play"

© NCERT not to be republished

Reprint 2026-27

7. Write a Python program to open the file *hello.txt* used in question no 6 in read mode to display its contents. What will be the difference if the file was opened in write mode instead of append mode?

8. Write a program to accept string/sentences from the user till the user enters "END" to. Save the data in a text file and then display only those sentences which begin with an uppercase alphabet.

9. Define pickling in Python. Explain serialization and deserialization of Python object.

10. Write a program to enter the following records in a binary file:

Item No	integer
Item_Name	string
Qty	integer
Price	float

Number of records to be entered should be accepted from the user. Read the file to display the records in the following format:

Item No:
Item Name :
Quantity:
Price per item:
Amount: (to be calculated as Price * Qty)

© NCERT not to be republished

Reprint 2026-27

Chapter 3

Stack

12130CH03

"We're going to be able to ask our computers to monitor things for us, and when certain conditions happen, are triggered, the computers will take certain actions and inform us after the fact."

— *Steve Jobs*

In this Chapter

- *Introduction*
- *Stack*
- *Operations on Stack*
- *Implementation of Stack in Python*
- *Notations for Arithmetic Expressions*
- *Conversion From Infix To Postfix Notation*
- *Evaluation of Postfix Expression*

3.1 Introduction

We have learnt about different data types in Python for handling values in Class XI. Recall that String, List, Set, Tuple, etc. are the sequence data types that can be used to represent collection of elements either of the same type or different types. Multiple data elements are grouped in a particular way for faster accessibility and efficient storage of data. That is why we have used different data types in python for storing data values. Such grouping is referred as a data structure.

A data structure defines a mechanism to store, organise and access data along with operations (processing) that can be efficiently performed on the data. For example, string is a data structure containing a sequence of elements where each element is a character. On the other hand, list is a sequence data structure in which each element may be of different types. We can apply different operations like reversal, slicing, counting of

© NCERT not to be republished

elements, etc. on list and string. Hence, a data structure organises multiple elements in a way so that certain operations on each element as well as the collective data unit could be performed easily.

Stack and *Queue* are two other popular data structures used in programming. Although not directly available in Python, it is important to learn these concepts as they are extensively used in a number of programming languages. In this chapter, we will study about stack, its implementation using Python as well as its applications.

Other important data structures in Computer Science include Array, Linked List, Binary Trees, Heaps, Graph, Sparse Matrix, etc.

3.2 Stack

We have seen piles of books in the library or stack of plates at home (Figure 3.1). To put another book or another plate in such a pile, we always place (add to the pile) the object at the top only. Likewise, to remove a book or a plate from such a pile, we always remove (delete from the pile) the object from the top only. This is because in a large pile, it is inconvenient to add or remove an object from in between or bottom. Such an arrangement of elements in a linear order is called a stack. We add new elements or remove existing elements from the same end, commonly referred to as the *top* of the stack. It thus follows the Last-In-First-out (LIFO) principle. That is, the element which was inserted last (the most recent element) will be the first one to be taken out from the stack.

A data structure in which elements are organised in a sequence is called linear data structure.

Figure 3.1: Stack of plates and books

© NCERT not to be republished

3.2.1 APPLICATIONS OF STACK

Some of the applications of stack in real-life are:

- Pile of clothes in an almirah
- Multiple chairs in a vertical pile
- Bangles worn on wrist
- Pile of boxes of eatables in pantry or on a kitchen shelf

Some examples of application of stack in programming are as follows:

- When we need to reverse a string, the string is traversed from the last character till the first character. i.e. characters are traversed in the reverse order of their appearance in the string. This is very easily done by putting the characters of a string in a stack.
- We use text/image editor for editing the text/image where we have options to redo/undo the editing done. When we click on the redo /undo icon, the most recent editing is redone/undone. In this scenario, the system uses a stack to keep track of changes made.
- While browsing the web, we move from one web page to another by accessing links between them. In order to go back to the last visited web page, we may use the back button on the browser. Let us say we accessed a web page P1 from where we moved to web page P2 followed by browsing of web page P3. Currently, we are on web page P3 and want to revisit web page P1. We may go to a previously visited web page by using the BACK button of the browser. On clicking the BACK button once, we are taken from web page P3 to web page P2, another click on BACK shows web page P1. In this case, the history of browsed pages is maintained as stack.
- While writing any arithmetic expression in a program, we may use parentheses to order the evaluation of operators. While executing the program, the compiler checks for matched parentheses i.e. each opening parenthesis should have a corresponding closing parenthesis and the pairs of parentheses are properly nested. In case of parentheses are

Think and Reflect

How does a compiler or an interpreter handle function calls in a program?

Think and Reflect

The operating system in computer or mobile allocates memory to different applications for their execution. How does an operating system keep track of the free memory that can be allocated among programs/ applications to be executed?

© NCERT not to be republished

Reprint 2026-27

mismatched, the compiler needs to throw an error. To handle matching of parentheses, stack is used.

3.3 Operations on Stack

As explained in the previous section, a stack is a mechanism that implements LIFO arrangement hence elements are added and deleted from the stack at one end only. The end from which elements are added or deleted is called TOP of the stack. Two fundamental operations performed on the stack are PUSH and POP. In this section, we will learn about them and implement them using Python.

3.3.1 PUSH and POP Operations

- `PUSH` adds a new element at the TOP of the stack. It is an insertion operation. We can add elements to a stack until it is full. A stack is full when no more elements can be added to it. Trying to add an element to a full stack results in an exception called 'overflow'.
- `POP` operation is used to remove the top most element of the stack, that is, the element at the TOP of the stack. It is a delete operation. We can delete elements from a stack until it is empty i.e. there is no element in it. Trying to delete an element from an empty stack results in an exception called 'underflow'.

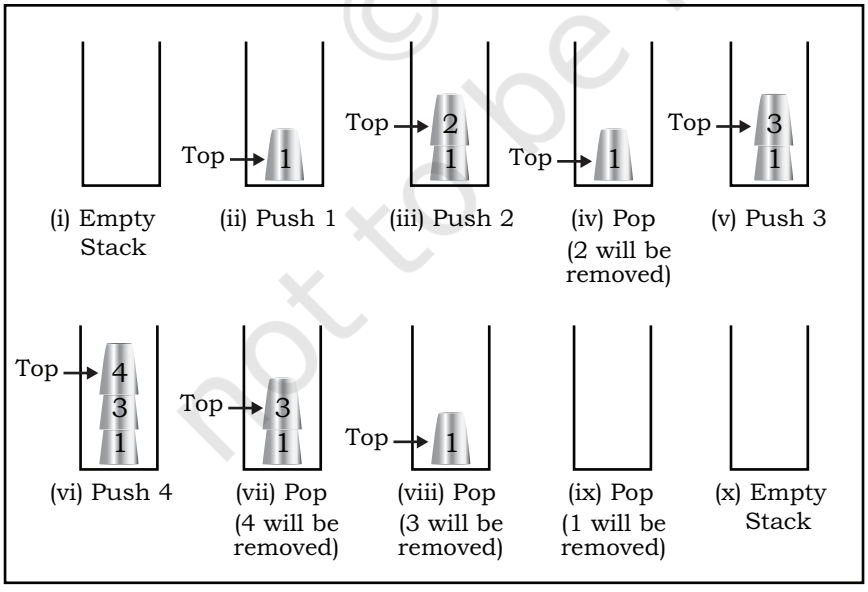


Figure 3.2: PUSH and POP operations on the stack of glasses

A stack is used to insert and delete elements in LIFO order. Same principle is followed in adding and removing glasses from a pile of glasses. Let us create a stack of glasses assuming that each glass is numbered. Visual representations of PUSH and POP operations on a stack of glasses are shown in Figure 3.2.

© NCERT not to be republished

3.4 Implementation of Stack in Python

We have learnt so far that a stack is a linear and ordered collection of elements. The simple way to implement a stack in Python is using the data type *list*. We can fix either of the sides of the list as TOP to insert/remove elements. It is to be noted that we are using built-in methods append() and pop() of the list for implementation of the stack. As these built-in methods insert/delete elements at the rightmost end of the list, hence explicit declaration of TOP is not needed.

Let us write a program to create a STACK (stack of glasses as given in Figure 3.2) in which we will:

- insert/delete elements (glasses)
- check if the STACK is empty (no glasses in the stack)
- find the number of elements (glasses) in the STACK
- read the value of the topmost element (number on the topmost glass) in the STACK

The program shall define the following functions to perform these operations:

- Let us create an empty stack named `glassStack`. We will do so by assigning an empty list to the identifier named `glassStack`:

```
glassStack = list()
```

- A function named isEmpty to check whether the stack `glassStack` is empty or not. Remember trying to remove an element from an empty stack would result in 'underflow'. This function returns True if the stack is empty, else returns False.

```
def isEmpty(glassStack):
    if len(glassStack)==0:
        return True
    else:
        return False
```

- A function named opPush to insert (PUSH) a new element in stack. This function has two parameters - the name of the stack in which the element is to be inserted (`glassStack`) and the element that needs to be inserted. We know that insertion of an element is always done at the TOP of the stack. Hence, we

© NCERT not to be republished

shall use the built-in method `append()` of list to add an element to the stack that always adds at the end of the list. As there is no limit on the size of list in Python, the implemented stack will never be full unless there is no more space available in memory. Hence, we will never face 'overflow' (no space for new element) condition for stack.

```
def opPush(glassStack,element):
    glassStack.append(element)
```

- A function named `size` to read the number of elements in the `glassStack`. We will use the len() function of list in Python to find the number of elements in the `glassStack`.

```
def size(glassStack):
    return len(glassStack)
```

- A function named top to read the most recent element (`TOP`) in the `glassStack`.

```
def top(glassStack):
    if isEmpty(glassStack):
        print('Stack is empty')
        return None
    else:
        x =len(glassStack)
        element=glassStack[x-1]
        return element
```

- A function named `opPop` to delete the topmost element from the stack. It takes one parameter - the name of the stack (`glassStack`) from which element is to be deleted and returns the value of the deleted element. The function first checks whether the stack is empty or not. If it is not empty, it removes the topmost element from it. We shall use the built-in method `pop()` of Python list that removes the element from the end of the list.

```
def opPop(glassStack):
    if isEmpty(glassStack):
        print('underflow')
        return None
    else:
        return(glassStack.pop())
```

© NCERT not to be republished

- A function named display to show the contents of the stack.

```
def display(glassStack):
    x=len(glassStack)
    print("Current elements in the stack
are: ")
    for i in range(x-1,-1,-1):
        print(glassStack[i])
```

Once we define the above functions we can use the following Python code to implement a stack of glasses.

```
glassStack = list()  # create empty stack

#add elements to stack
element='glass1'
print("Pushing element ",element)
opPush(glassStack,element)
element='glass2'
print("Pushing element ",element)
opPush(glassStack,element)

#display number of elements in stack
print("Current number of elements in stack
is",size(glassStack))

#delete an element from the stack
element=opPop(glassStack)
print("Popped element is",element)

#add new element to stack
element='glass3'
print("Pushing element ",element)
opPush(glassStack,element)

#display the last element added to the
#stack
print("top element is",top(glassStack))

#display all elements in the stack
display(glassStack)
```

© NCERT not to be republished

```
#delete all elements from stack
while True:
   item=opPop(glassStack)
   if item == None:
      print("Stack is empty now")
      break
   else:
      print("Popped element is",item)
```

```
The output of the above program will be as
follows:
Pushing element  glass1
Pushing element  glass2
Current number of elements in stack is 2
Popped element is glass2
Pushing element  glass3
top element is glass3
Current elements in the stack are:
glass3
glass1
Popped element is glass3
Popped element is glass1
Underflow
Stack is empty now
```

3.5 NOTATIONS FOR ARITHMETIC EXPRESSIONS

We write arithmetic expressions using operators in between operands, like x + y, 2 - 3 * y, etc. and use parentheses () to order the evaluation of operators in complex expressions. These expressions follow infix representation and are evaluated using BODMAS rule.

Polish mathematician Jan Lukasiewicz in the 1920's introduced a different way of representing arithmetic expression, called polish notation. In such notation, operators are written before their operands. So the order of operations and operands determines the result, making parentheses unnecessary. For example, we can write x+y in polish notation as +xy. This is also called prefix notation as we prefix the operator before operands.

By reversing this logic, we can write an expression by putting operators after their operands. For example, x+y can be written as xy+. This is called reverse polish

© NCERT not to be republished

notation or postfix notation. To summarise, any arithmetic expression can be represented in any of the three notations viz. Infix, Prefix and Postfix and are listed in Table 3.1 with examples.

Table 3.1 Infix, Prefix and Postfix Notations

Type of Expression	Description	Example
Infix	Operators are placed in between the operands	x * y + z 3 *(4 + 5) (x + y)/(z * 5)
Prefix (Polish)	Operators are placed before the corresponding operands	+z*xy *3+45 /+xy*z5
Postfix (Reverse Polish)	Operators are placed after the corresponding operands	xy*z+ 345+* xy+z5*/

3.6 Conversion from Infix to Postfix Notation

It is easy for humans to evaluate an *infix* expression. Consider an *infix* expression x + y / z. While going from left to right we first encounter + operator, but we do not add x + y and rather evaluate y/z, followed by addition operation. This is because we know the order of precedence of operators that follows BODMAS rule. But, how do we pass this precedence knowledge to the computer through an expression?

In contrast, *prefix/postfix* expressions do not have to deal with such precedence because the operators are already positioned according to their order of evaluation. Hence a single traversal from left to right is sufficient to evaluate the expression. In this section, we will learn about the conversion of an arithmetic expression written in *infix* notation to its equivalent expression in *postfix* notation using a stack.

During such conversion, a stack is used to keep track of the operators encountered in the *infix* expression. A variable of string type is used to store the equivalent *postfix* expression. Algorithm 3.1 converts an expression in *infix* notation to *postfix* notation:

Think and Reflect

Write an algorithm to convert an infix expression into equivalent prefix expression using stack.

Algorithm 3.1: Conversion of expression from infix to postfix notation

Step 1: Create an empty string named postExp to store the converted postfix expression.
Step 2: INPUT infix expression in a variable, say inExp
Step 3: For each character in inExp, REPEAT Step 4

© NCERT not to be republished

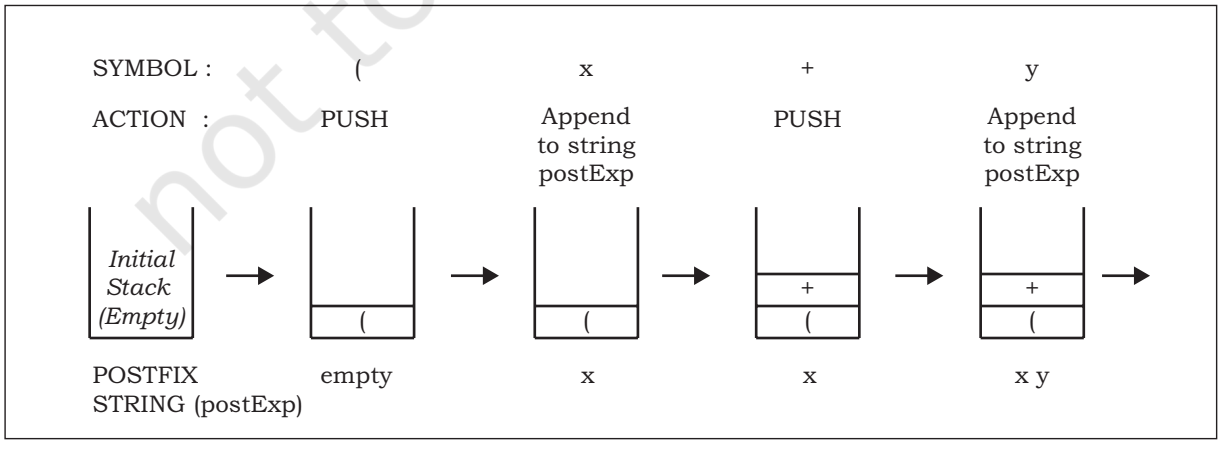
Step 4: IF character is a left parenthesis THEN PUSH on the Stack
ELSE IF character is a right parenthesis
THEN POP the elements from the Stack and append to string postExp until the corresponding left parenthesis is popped while discarding both left and right parentheses
ELSE IF character is an operator
THEN IF its precedence is lower than that of operator at the top of Stack
THEN POP elements from the Stack till an operator with precedence less than the current operator is encountered and append to string postExp before pushing this operator on the postStack
ELSE PUSH operator on the Stack
ELSE Append the character to postExp

Step 5: Pop elements from the Stack and append to postExp until Stack is empty

Step 6: OUTPUT postExp

Example 3.1

Let us now use this algorithm to convert a given infix expression (x + y) / (z*8) into equivalent postfix expression using a stack. Figure 3.3 shows the steps to be followed on encountering an operator or an operand in the given infix expression. Note here that stack is used to track the operators and parentheses, and a string variable contains the equivalent postfix expression. Initially both are empty. Each character in the given infix expression is processed from left to right and the appropriate action is taken as detailed in the algorithm. When each character in the given infix expression has been processed, the string will contain the equivalent postfix expression.



© NCERT not to be republished

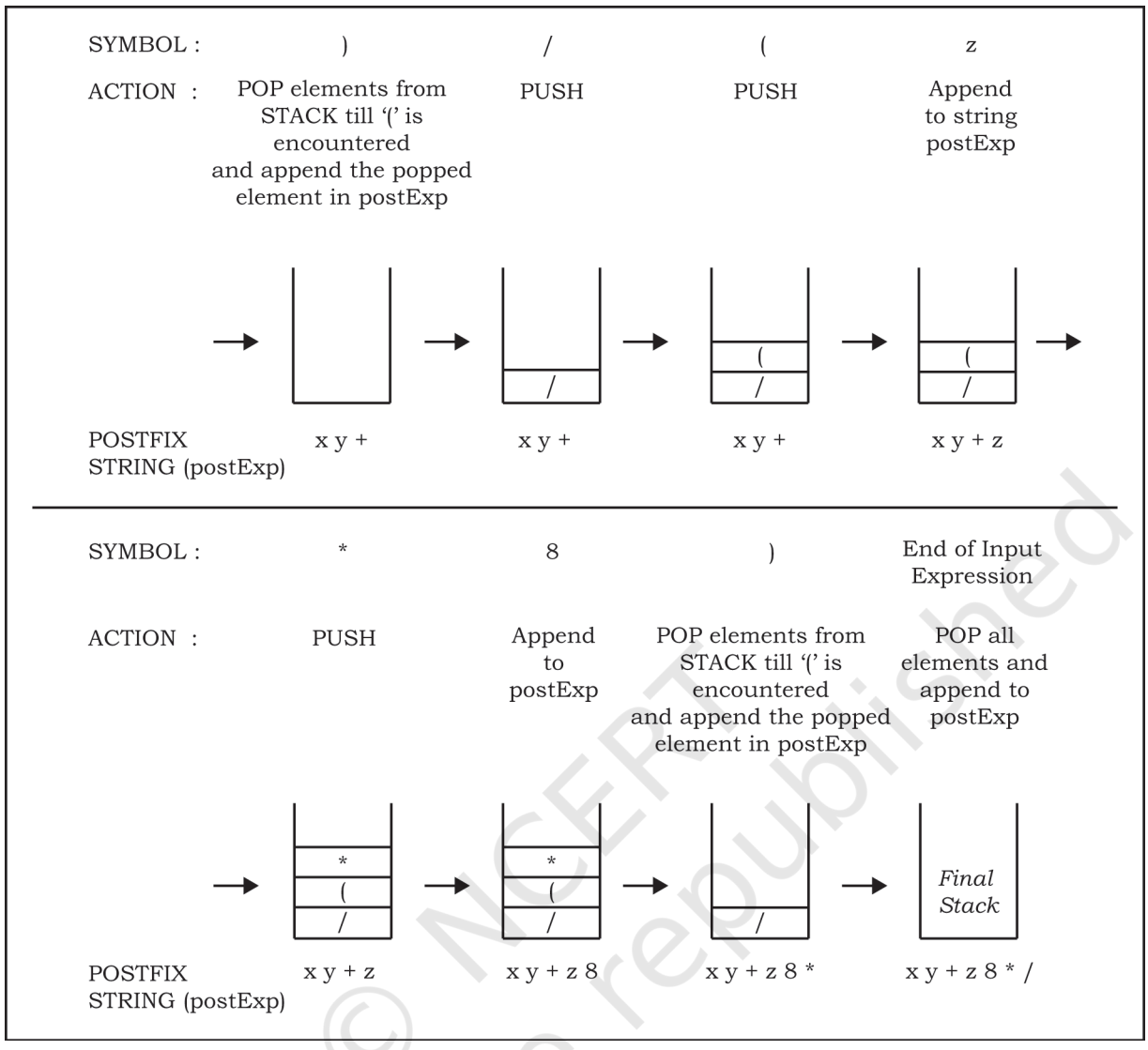


*Figure 3.3: Conversion of infix expression (x + y)/(z*8) to postfix notation*

3.7 Evaluation of Postfix Expression

Stacks can be used to evaluate an expression in postfix notation. For simplification, we are assuming that operators used in expressions are binary operators. The detailed step by step procedure is given in Algorithm 3.2.

Algorithm 3.2: Evaluation of postfix expression

Step 1: INPUT postfix expression in a variable, say postExp

Step 2: For each character in postExp, REPEAT Step 3

Step 3: IF character is an operand

THEN PUSH character on the Stack

ELSE POP two elements from the Stack, apply the operator on

© NCERT not to be republished

the popped elements and PUSH the computed value onto the Stack

Step 4: IF Stack has a single element

THEN POP the element and OUTPUT as the net result

ELSE OUTPUT "Invaild Postfix expression"

Example 3.2

Figure 3.4 shows the step-by-step process of evaluation of the postfix expression 7 8 2 * 4 / + using Algorithm 3.2 .

Think and Reflect

Write an algorithm to evaluate any prefix expression using a stack.

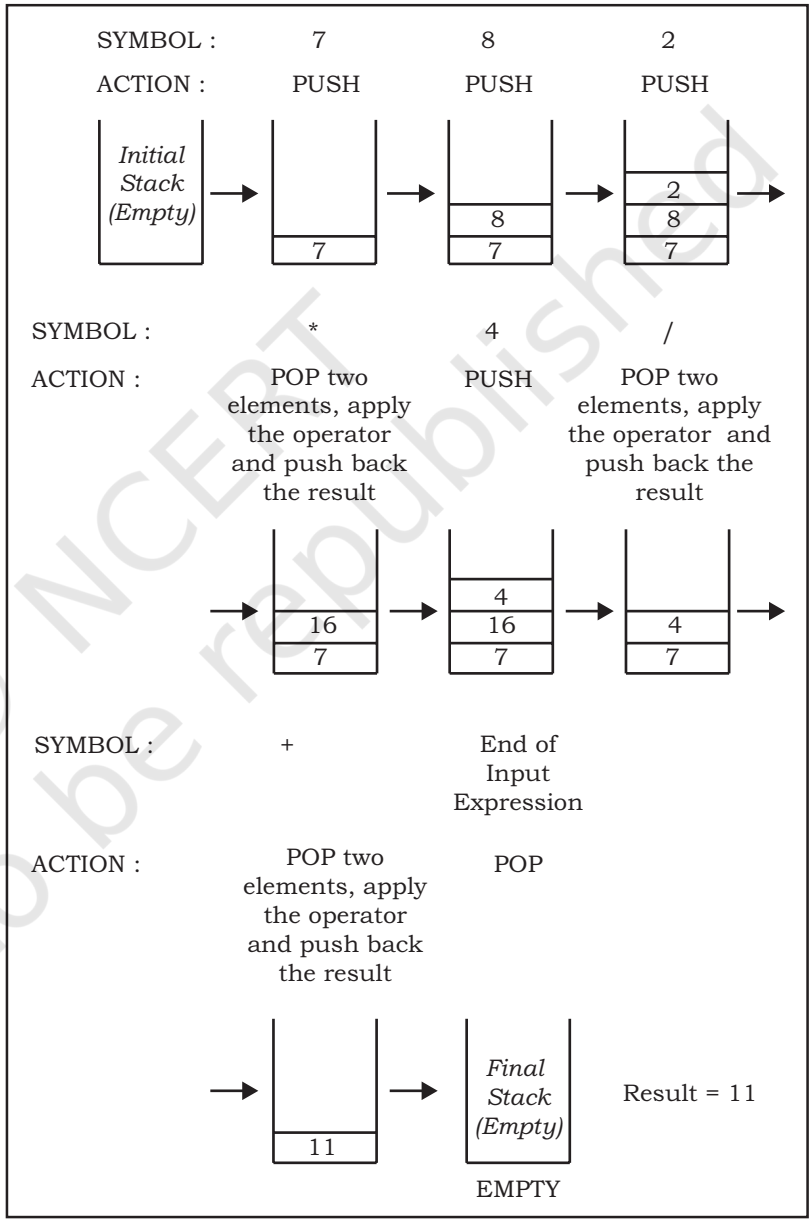


*Figure 3.4: Evaluation of postfix expression 7 8 2 * 4 / +*

© NCERT not to be republished

Reprint 2026-27

SUMMARY

- Stack is a data structure in which insertion and deletion is done from one end only, usually referred to as TOP.
- Stack follows LIFO principle using which an element inserted in the last will be the first one to be out.
- PUSH and POP are two basic operations performed on a stack for insertion and deletion of elements, respectively.
- Trying to pop an element from an empty stack results into a special condition *underflow.*
- In Python, list is used for implementing a stack and its built-in-functions *append* and *pop* are used for insertion and deletion, respectively. Hence, no explicit declaration of TOP is needed.
- Any arithmetic expression can be represented in any of the three notations viz. Infix, Prefix and Postfix.
- While programming, Infix notation is used for writing an expression in which binary operators are written in between the operands.
- A single traversal from left to right of Prefix/ Postfix expression is sufficient to evaluate the expression as operators are correctly placed as per their order of precedence.
- Stack is commonly used data structure to convert an Infix expression into equivalent Prefix/Postfix notation.
- While conversion of an Infix notation to its equivalent Prefix/Postfix notation, only operators are PUSHed onto the Stack.
- When evaluating any Postfix expression using Stack, only operands are PUSHed onto it.

© NCERT not to be republished

Reprint 2026-27

EXERCISE

1. State TRUE or FALSE for the following cases:
   a) Stack is a linear data structure
   b) Stack does not follow LIFO rule
   c) PUSH operation may result into underflow condition
   d) In POSTFIX notation for expression, operators are placed after operands

2. Find the output of the following code:
   a)
```
result=0
numberList=[10,20,30]
numberList.append(40)
result=result+numberList.pop()
result=result+numberList.pop()
print("Result=",result)
```
   b)
```
answer=[]; output=''
answer.append('T')
answer.append('A')
answer.append('M')
ch=answer.pop()
output=output+ch
ch=answer.pop()
output=output+ch
ch=answer.pop()
output=output+ch
print("Result=",output)
```

3. Write a program to reverse a string using stack.

4. For the following arithmetic expression:
   ((2+3)*(4/2))+2
   Show step-by-step process for matching parentheses using stack data structure.

5. Evaluate following postfix expressions while showing status of stack after each operation given A=3, B=5, C=1, D=4
   a) A B + C *
   b) A B * C / D *

6. Convert the following infix notations to postfix notations, showing stack and string contents at each step.
   a) A + B - C * D
   b) A * ((C + D)/E)

7. Write a program to create a Stack for storing only odd numbers out of all the numbers entered by the user. Display the content of the Stack along with the largest odd number in the Stack. (Hint. Keep popping out the elements from stack and maintain the largest element retrieved so far in a variable. Repeat till Stack is empty)

© NCERT not to be republished

Chapter 4

Queue

12130CH04

"We could say we want the Web to reflect a vision of the world where everything is done democratically. To do that, we get computers to talk with each other in such a way as to promote that ideal."

— Tim Berners-Lee

In this Chapter

- *Introduction to Queue*
- *Operations on Queue*
- *Implementation of Queue using Python*
- *Introduction to Deque*
- *Implementation of Deque using Python*

4.1 Introduction to Queue

In the previous chapter we learned about a data structure called Stack, which works on Last-In-First-Out (LIFO) principle. In this chapter, we will learn about another data structure called Queue which works on First-In-First-Out (FIFO) principle. Queue is an ordered linear list of elements, having different ends for adding and removing elements in it.

Figure 4.1: Queue of people at a bank

Examples of queue in our everyday life include students standing in a queue for morning assembly, customers forming a queue at the cash counter in a bank (Figure 4.1), vehicles queued at fuel pumps (Figure 4.2), etc.

© NCERT not to be republished

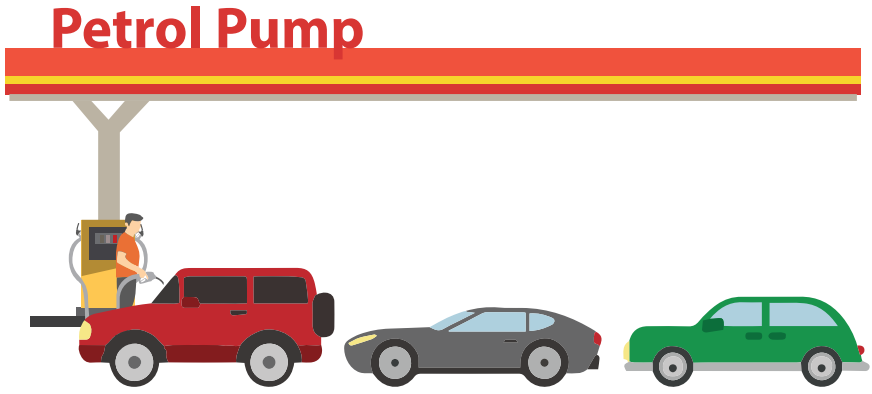


Figure 4.2: Queue of cars in a petrol pump

4.1.1 First In First Out (FIFO)

Queue follows the principle of First In First Out (FIFO), since the element entering first in the queue will be the first one to come out of it. Thus, the element that has been longest in the queue will be removed first. It is also known as a First Come First Served (FCFS) approach. Queue is an arrangement in which new objects/items always get added at one end, usually called the REAR, and objects/items always get removed from the other end, usually called the FRONT of the queue. REAR is also known as TAIL and FRONT as HEAD of a queue.

4.1.2 Applications of Queue

(A) The concept of queue has many applications in real-life:

- If a train ticket is in the waiting list (such as W/L1), it means the ticket is in a queue of tickets waiting to get confirmed, as per the increasing order of waiting numbers. If a confirmed ticket is cancelled, the W/L1 numbered ticket is removed from the FRONT of the waiting queue and confirmed.
- Sometimes on calling a customer service centre, the Interactive Voice Response System (IVRS) tells us to wait till a support person is available. Here the call is put into a queue of customers waiting to be serviced.
- Imagine there is a single-lane one-way road, then the vehicle that entered first will exit first, following the concept of queue. Likewise, vehicles in a highway toll tax booth are served following the principle of FIFO.

© NCERT not to be republished

(B) Following are some examples of application of queue in computer science:

- Suppose there is a web-server hosting a web-site to declare result(s). This server can handle a maximum of 50 concurrent requests to view result(s). So, to serve thousands of user requests, a Queue would be the most appropriate data structure to use.
- Some Operating Systems (OS) are required to handle multiple tasks called - jobs, seeking to use the processor. But we know that a processor can handle only one task at a time. Therefore, in a multitasking operating system, jobs are lined up (queued) and then given access to the processor according to some order. The simplest way is to give access to the processor on a FIFO basis, that is according to the order in which the jobs arrive with a request for the processor.
- When we send print commands from multiple files from the same computer or from different computers using a shared printer. The OS puts these print requests in a queue and sends them to the printer one by one on a FIFO basis.

Think and Reflect

In the web-server example (for result declaration), suppose the server receives a request from an Administrator to access the result of a school on an urgent basis, along with other requests from students to check individual results. Can you suggest some strategy to ensure service to all as per their urgency?

4.2 Operations on Queue

Following the FIFO approach, data structure queue supports the following operations:

- ENQUEUE: is used to insert a new element to the queue at the rear end. We can insert elements in the queue till there is space in the queue for adding more elements. Inserting elements beyond capacity of the queue will result in an exception - known as Overflow.
- DEQUEUE: is used to remove one element at a time from the front of the queue. We can delete elements from a queue until it is empty, trying to delete an element from an empty queue will result in exception - known as Underflow.

To perform enqueue and dequeue efficiently on a queue, following operations are also required:

- `IS EMPTY` : used to check whether the queue has any element or not, so as to avoid Underflow exception while performing dequeue operation.

© NCERT not to be republished

Reprint 2026-27

While using a list to implement queue, we can designate either end of the list as Front or Rear of the queue. But we have to fix either of the ends index[0] or index[n-1] as Front and fix the opposite end as Rear.

- PEEK : used to view elements at the front of the queue, without removing it from the queue.
- IS FULL : used to check whether any more elements can be added to the queue or not, to avoid Overflow exceptions while performing enqueue operation.

Figure 4.3 shows the various stages of a simple queue containing alphabets. In the figure, Front of the queue is on the left and Rear on the right.

Operation performed	Status of queue after operation
enqueue(z)	F→ Z ←R
enqueue(x)	F→ Z X ←R
enqueue(c)	F→ Z X C ←R
dequeue()	F→ X C ←R
enqueue(v)	F→ X C V ←R
dequeue()	F→ C V ←R
dequeue()	F→ V ←R

Figure 4.3: Various Stages of Stack Operations

4.3 Implementation of Queue using Python

There are many ways in which queues can be implemented in a computer program, one way is using the list data type of Python. For creating a queue structure in the program, following functions need to be defined:

- Let's create a queue named myQueue. We can create it by assigning an empty list.

```
myQueue = list()
```

- A function (enqueue) to insert a new element at the end of queue. The function has two parameters - name of the queue and element which is to be inserted in the queue.

```
def enqueue(myQueue, element):
    myQueue.append(element)
```

Note: append() function always adds an element at the end of the list, hence Rear of queue.

- We don't need to implement Is Full, as Python being a dynamic language, does not ask for the

© NCERT not to be republished

creation of list having fixed size. Hence, we will never encounter a situation when the queue is full.

- A function (`isEmpty`) to check, if the queue has an element or not? This can be done by checking the length of the queue. The function has a parameter -- name of the queue and returns True if the queue is empty False otherwise.

```
def isEmpty(myQueue):
      if len(myQueue)==0:
            return True
      else:
            return False
```

- A function (`dequeue`) to delete an element from the front of the queue. It has one parameter - name of the queue and returns the deleted element. The function first checks if the queue is empty or not, for successful deletion.

```
def dequeue(myQueue):
      if not (isEmpty(myQueue)):
            return myQueue.pop(0)
      else :
            print("Queue is empty")
```

Note: The pop() function with index[0] will delete the element from the beginning of the list, hence Front of queue.

- A function (size) to get the number of elements in the queue. We can use the len() function of Python's list to find the number of elements in the queue. The function has one parameter - name of the queue and returns the number of elements in the queue.

```
def size(myQueue):
    return len(myQueue)
```

- A function (peek) to simply read, but not to delete, the element at the front end of the queue. For this, we can read the element at index[0] of the queue. The function has one parameter - name of the queue and returns the value of element at Front if queue is not empty, None otherwise.

```
def peek(myQueue):
    if isEmpty(myQueue):
          print('Queue is empty')
          return None
    else:
          return myQueue[0]
```

Think and Reflect

Can you implement a queue data structure using tuple or dictionary?

While choosing the name of above functions general naming convention w.r.t. the queue is followed. As these are user defined functions any other name can also be used.

© NCERT not to be republished

Reprint 2026-27

Activity 4.1

How can you avoid printing of None, when trying to print an empty queue?

Activity 4.2

What if the content of the complete queue is to be listed? Write a function for it.

Let us consider the example of a queue that people form while waiting at a bank cash counter. Usually, following are the events that occur in queue:

- Two friends come together and go to the cash counter, i.e. they form a queue - enqueue operation is performed two times.
- As soon as the person at the front is serviced, he will be removed from the queue - thus dequeue operation is performed. Cashier calls Next to serve the next person who is now at the front of the queue.
- Cashier wants to know the length of the queue - size of the queue is checked.
- Meanwhile, a few more people walk in the bank, and three of them join the queue at the cash counter, i.e. enqueue happens 3 times.
- Another person gets served and leaves the counter, i.e. dequeue is performed. Cashier calls Next to serve another person.
- The Next three people get served one after another, i.e. dequeue is performed thrice.
- Cashier calls Next and realises that there are no more people to be served - underflow situation happens

Now, let us write the code for the above scenario of the bank.

Program 4-1

```
myQueue = list()
# each person to be assigned a code as P1, P2, P3,...
element = input("enter person's code to enter in queue :")
enqueue(myQueue,element)
element = input("enter person's code for insertion in queue :")
enqueue(myQueue,element)
print("person removed from queue is:", dequeue(myQueue))
print("Number of people in the queue is :",size(myQueue))
element = input("enter person's code to enter in queue :")
enqueue(myQueue,element)
element = input("enter person's code to enter in queue :")
enqueue(myQueue,element)
element = input("enter person's code to enter in queue :")
enqueue(myQueue,element)
```

© NCERT not to be republished

```
print("Now we are going to remove remaining people from the
queue")
while not isEmpty(myQueue):
      print("person removed from queue is ",
      dequeue(myQueue))
```

Output

```
enter person's code to enter in queue :P1
enter person's code to enter in queue :P2
person removed from the queue is :p1
number of people in the queue is :1
enter person's code to enter in queue :P3
enter person's code to enter in queue :P4
enter person's code to enter in queue :P5
Now we are going to remove remaining people from the queue
person removed from the queue is :p2
person removed from the queue is :p3
person removed from the queue is :p4
person removed from the queue is :p5
Queue is empty
```

4.4 Introduction to Deque

Deque (pronounced as "deck") is an arrangement in which addition and removal of element(s) can happen from any end, i.e. head/front or tail/rear. This data structure does not apply any restriction on the side from which addition/removal of elements should happen, so it can be used to implement stack or queue in the program. It is also known as Double ended queue, because it permits insertion, deletion operations from any end.

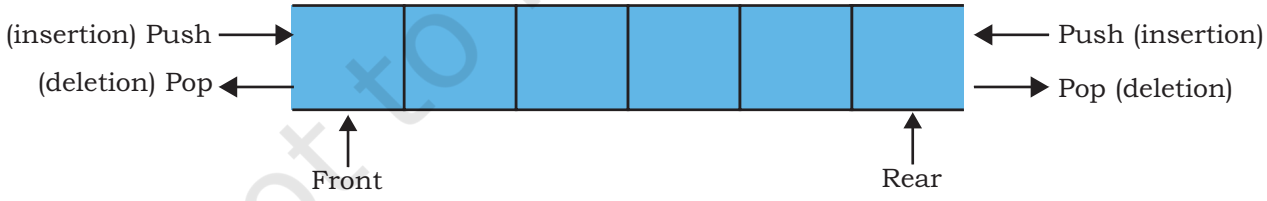


Figure 4.4: Basic deque structure displaying head and tail to implement stack or queue.

4.4.1 Applications of Deque

- At a train ticket purchasing counter, a normal queue of people is formed for purchasing a ticket. A person at the front purchased the ticket and left the counter. After a while they return back to the counter to ask

© NCERT not to be republished

something. As they have already purchased a ticket, they may have the privilege to join the queue from the front.

- Vehicles in a highway toll tax booth are served following the principle of queue. There are multiple queues if there are parallel booths at the toll gate. In case all vehicles of a booth are served then vehicles from the other booth(s) are asked to form a queue in front of the vacant booth. So, vehicles at the end of those queues will leave (removed from the end from where queue was joined) current booth and join queue at the vacant booth.

Following are some examples where data structure deque maybe applied in computer science:

- To maintain browser history (URL), usually a stack is used, because once a tab is closed and if you press ctrl+shift+T, the most recently closed URL is opened first. As the number of URLs which can be stored in history is fixed, so when this list of URLs becomes large, URLs from the end of the list (i.e. which were least visited) gets deleted.
- Same happens for providing the Do and Undo option in any text editor.
- To check whether a given string is palindrome or not? Process string left to right (character wise) and insert it in deque from tail/rear like a normal queue. Once the entire string is processed (i.e. inserted in deque) we will take out (delete) a character from both the ends and match them till there is no character left or only one character left in deque. In either case, string is palindrome.

Activity 4.3

In a deque, if insertion and deletion of elements is done from the same end, it will behave as
1) Queue
2) Stack
3) List
4) None of the above

Activity 4.4

In a deque, if insertion and deletion of elements is done from the opposite end, it will behave as
1) Queue
2) Stack
3) List
4) None of the above

4.4.2 Operations on Deque

- INSERTFRONT: This operation is used to insert a new element at the front of the deque.
- INSERTREAR: This operation is the same as a normal queue, i.e. insert a new element at the rear of the deque.
- DELETIONFRONT: This operation is the same as normal queue, i.e. to remove an element from the front of the deque.
- DELETIONREAR: This operation is used to remove one element at a time from the rear of the deque.

© NCERT not to be republished

To perform above operations efficiently on a deque, we will need all supporting operations used in normal queue viz Is Empty, Peek, Size.

Let's understand how these operations work for checking whether a string is palindrome or not, using a deque through the following algorithm.

Algorithm 4.1

Step1: Start traversing string (madam) from left side, a character at a time.

Step 2: Insert the character in deque as normal queue using INSERTREAR.

Step 3: Repeat Step 1 and Step 2 for all characters of string (madam)

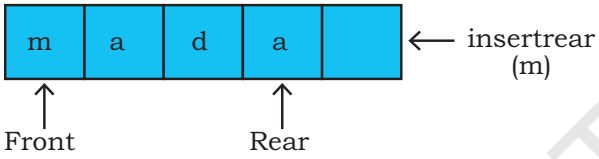


Figure 4.5: Status of Deque after 4th iteration

Step 4: Remove one character from the front and one character from the rear end of deque using DELETIONFRONT and DELETIONREAR we can do it.

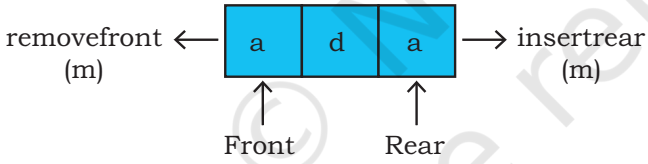


Figure 4.6: Status of Deque after removing one character from both the ends.

Step 5: Match these two removed characters.

Step 6: If they are same then

repeat Step 4 and 5 till deque is empty or left with only one character,

eventually string is Palindrome

else stop as string is not palindrome

4.5 Implementation of Deque Using Python

Like queue, deque is also an ordered linear list, hence we use list data type to create deque in our program. The program should have the following functions/ statement(s) defined in it:

© NCERT not to be republished

- A statement to create deque, with name myDeque.

```
myDeque = list()
```

- A function insertFront(), to insert an element at the front of deque having two parameters - name of deque and element to be inserted. As the element is to be inserted in the beginning, we will use insert() with index 0 for it.

```
def  insertFront(myDeque, element):
     myDeque.insert(0,element)
```

- A function insertRear(), to insert an element at the rear of deque. It's implementation will be the same as enqueue() of normal queue requiring two parameters same as insertFront().
- A function isEmpty(), to check the presence of element(s) in deque will be the same as the function, with the same name, defined for normal queue.
- A function deletionRear(), to delete an element from the rear of the deque. It only requires the name of deque and returns the deleted element. We will use pop() without parameter(s) to delete the last element of the deque.

```
def deletionRear(myDeque):
    if not (isEmpty()):
       return myDeque.pop()
      # removing data from end of list
    else :
       print("Deque empty")
```

- A function deletionFront(), to delete an element from the front of deque. It's implementation will be the same as dequeue() of normal queue.
- A function getFront(), to read value from the front of deque, without removing it from the queue when the queue is not empty. It accepts the name of deque as parameter and returns a copy of value.

```
def  getFront(mydeque):
    if not (isEmpty()):
       return mydeque[0]
    else :
      print(" Queue empty")
```

- A function getRear(), to read value from the rear of the deque, without removing it from the deque. The

© NCERT not to be republished

function accepts deque as argument and returns a copy of value, when the queue is not empty.

```
def  getRear(mydeque):
    if not (isempty()):
       return mydeque[len(mydeque)-1]
    else :
       print(" Deque empty")
```

Let us write a main(), function to invoke various Deque functions :

Program 4-2 Implementation of Deque in Python

```
def insertFront(myDeque,element):
     myDeque.insert(0,element)

def getFront(myDeque):
     if not (isEmpty(myDeque)):
          return myDeque[0]
     else:
          print("Queue underflow")

def getRear(myDeque):
     if not (isEmpty(myDeque)):
          return myDeque[len(myDeque)-1]
     else:
          print ("Queue underflow")

def insertRear(myDeque,element):
     myDeque.append(element)

def isEmpty(myDeque):
     if len(myDeque) == 0:
          return True
     else:
          return False

def deletionRear(myDeque):
     if not isEmpty(myDeque):
          return myDeque.pop()
     else:
          print("Queue underflow")

def deletionFront(myDeque):
     if isEmpty(myDeque):
```

© NCERT not to be republished

Reprint 2026-27

```
            print("Queue underflow")
      else:
            return myDeque.pop(0)

def main():
      dQu = list()
      choice = int(input('enter 1 to use as normal queue 2 otherwise
         : '))
      if choice == 1:
            element = input("data for insertion at rear ")
            insertRear(dQu,element)
            element = getFront(dQu)
            print("data at the beginning of queue is ", element)
            element = input("data for insertion at front ")
            insertRear(dQu,element)
            print('data removed from front of queue is ', deletionFront(dQu))
            print('data removed from front of queue is ', deletionFront(dQu))
```

Output

```
enter 1 to use as normal queue 2 otherwise : 1
data for insertion at rear    23
data at the beginning of queue is    23
data for insertion at rear   45
data removed from front of queue is    23
data removed from front of queue is    45
Queue underflow
data removed from front of queue is    None

enter 1 to use as normal queue 2 otherwise : 2
data for insertion at front    34
data at the end of queue is    34
data for insertion at front    56
data removed from rear of queue is    34
data removed from rear of queue is    56
Queue underflow
data removed from rear of queue is    None
```

SUMMARY

- Queue is an ordered linear data structure, following FIFO strategy.
- Front and Rear are used to indicate beginning and end of queue.
- In Python, the use of predefined methods takes care of Front and Rear.

© NCERT not to be republished

- Insertion in a queue happens at the rear end. Deletion happens at the front.
- Insertion operation is known as enqueue and deletion operation is known as dequeue.
- To support enqueue and dequeue operations, isEmpty, isfull and peek operations are used
- Deque is a version of queue, which allows insertion and deletion at both ends.
- A deque can support both stack and queue operations.
- Other operations supported by deque are insertfront, insertrear, deletefront, deleterear, getfront, getrear, isempty and isfull.

EXERCISE

1. Fill in the blank
   a) _____ is a linear list of elements in which insertion and deletion takes place from different ends.
   b) Operations on a queue are performed in _____ order.
   c) Insertion operation in a queue is called _____ and deletion operation in a queue is called _____.
   d) Deletion of elements is performed from _____ end of the queue.
   e) Elements 'A','S','D' and 'F' are present in the queue, and they are deleted one at a time, _____ is the sequence of element received.
   f) _____ is a data structure where elements can be added or removed at either end, but not in the middle.
   g) A deque contains 'z','x','c','v' and 'b' . Elements received after deletion are 'z','b','v','x' and 'c'. _____ _____ is the sequence of deletion operation performed on deque.
2. Compare and contrast queue with stack.
3. How does FIFO describe queue?

© NCERT not to be republished

Reprint 2026-27

4. Write a menu driven python program using queue, to implement movement of shuttlecock in it's box.

5. How is queue data type different from deque data type?

6. Show the status of queue after each operation

```
enqueue(34)
enqueue(54)
dequeue()
enqueue(12)
dequeue()
enqueue(61)
peek()
dequeue()
dequeue()
dequeue()
dequeue()
enqueue(1)
```

7. Show the status of deque after each operation

```
peek()
insertFront(12)
insertRear(67)
deletionFront()
insertRear(43)
deletionRear()
deletionFront()
deletionRear()
```

8. Write a python program to check whether the given string is palindrome or not, using deque. (Hint : refer to algorithm 4.1)

© NCERT not to be republished

Chapter 5

Sorting

12130CH05

"Every one of today's smartphones has thousands of times more processing power than the computers that guided astronauts to the moon."

— Peter Thiel

In this Chapter

- *Introduction*
- *Bubble Sort*
- *Selection Sort*
- *Insertion Sort*
- *Time Complexity of Algorithms*

5.1 Introduction

Sorting is the process of ordering or arranging a given collection of elements in some particular order. We can sort a collection of numbers in ascending (increasing) or descending (decreasing) order. If the collection is of strings, we can sort it in an alphabetical order (a-z or z-a) or according to the length of the string. For example, words in a dictionary are sorted in alphabetical order; seats in an examination hall are ordered according to candidates' roll number. We can also sort a list of students based on their height or weight.

Imagine finding the meaning of a word from a dictionary that is not ordered. We will have to search for the word on each page till we find the word, which will be very tedious. That is why dictionaries have the words in alphabetical order and it ease the process of searching.

© NCERT not to be republished

Think and Reflect

Can you identify other examples where sorting plays an important role in computers?

Sorting a large number of items can take a substantial amount of time. However, this extra time (called overhead) is worth when compared to the amount of time needed to find an element from an unsorted list. Sorting is an important area of study in computer science, and many sorting algorithms have been developed and analysed from their performance point of view. In this chapter, we will learn about three sorting methods and implement them using Python. Bubble sort is discussed in section 5.2, followed by discussion on selection sort and insertion sort in section 5.3 and 5.4, respectively

5.2 Bubble Sort

The first sorting technique we are going to understand is Bubble sort. It sorts a given list of elements by repeatedly comparing the adjacent elements and swapping them if they are unordered. Swapping two elements means changing their positions with each other. In algorithm, every iteration through each element of a list is called a pass. For a list with n elements, the bubble sort makes a total of n – 1 passes to sort the list. In each pass, the required pairs of adjacent elements of the list will be compared. In order to arrange elements in ascending order, the largest element is identified after each pass and placed at the correct position in the list. This can be considered as the largest element being 'bubbled up'. Hence the name Bubble sort. This sorted element is not considered in the remaining passes and thus the list of elements gets reduced in successive passes.

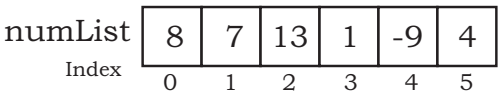
Think and Reflect

In Figure 5.1, we can see that the list got sorted in the 4th pass itself. Still the bubble sort technique made a redundant 5th pass which did not result in any swap. If there is no swapping in any pass, it means the list is already sorted, hence the sorting operation needs to be stopped. Can you think of making any improvement in the Algorithm 5.1 so that it stops when the list becomes sorted?

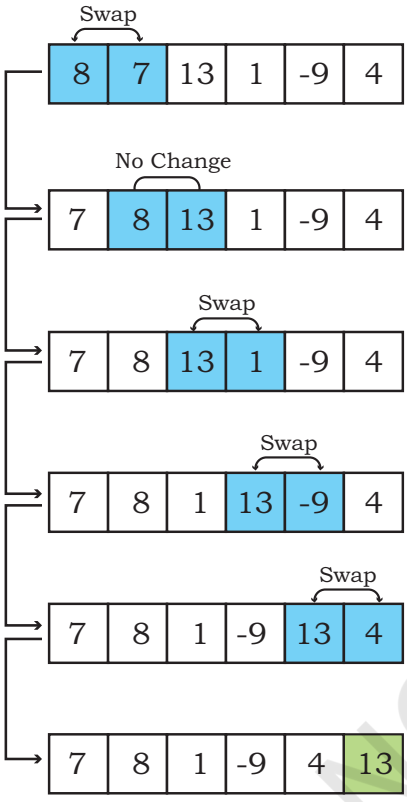
Figure 5.1 demonstrates the working of the bubble sort method to arrange a list in ascending order. Let us consider a list having 6 elements as numList = [8, 7, 13, 1, -9, 4]. In the figure, elements being compared are highlighted with blue colour and sorted elements are highlighted with green colour. To begin sorting, the element at index 0 is compared with the element at index 1. If the first element is bigger, it is swapped with the second. Else, no change is done. Next, the element at index 1 is compared with the element at index 2. This continues till the end of the list is reached. After the first pass, the largest element will reach the end of the list as shown in Figure 5.1 with green colour.

© NCERT not to be republished

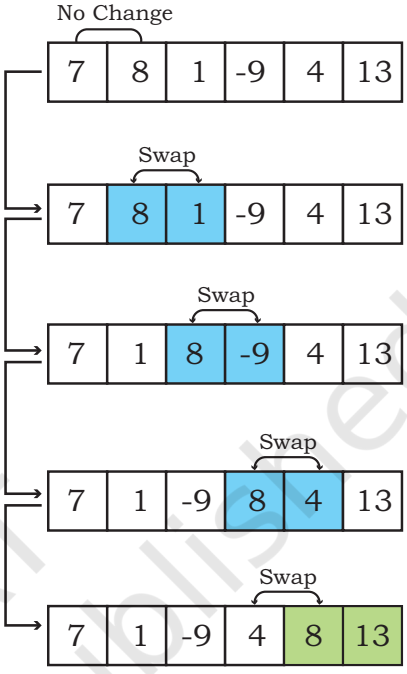
Reprint 2026-27



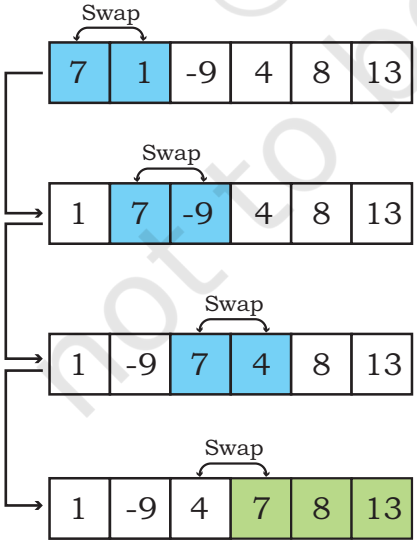
Comparison in Pass 1



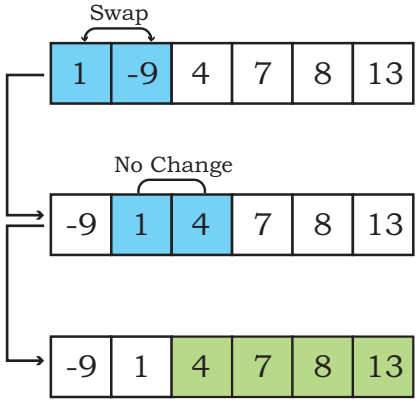
Comparison in Pass 2



Comparison in Pass 3



Comparison in Pass 4



© NCERT not to be republished

Figure 5.1: Comparisons done in different passes of Bubble sort

Algorithm 5.1 shows the steps followed for the bubble sort that takes `numList` as a list of n elements, and sorts the list in ascending order:

Activity 5.1

Algorithm 5.1 sorts a list in ascending order. Write a bubble sort algorithm to sort a list in descending order?

Algorithm 5.1: Bubble Sort

BUBBLESORT(numList, n)

Step 1: SET i = 0

Step 2: WHILE i< n REPEAT STEPS 3 to 8

Step 3: SET j = 0

Step 4: WHILE j< n-i-1,REPEAT STEPS 5 to 7

Step 5: IF numList[j] > numList[j+1] THEN

Step 6: swap(numList[j],numList[j+1])

Step 7: SET j=j+1

Step 8: SET i=i+1

Program 5-1 Implementation of bubble sort using Python.

```
def bubble_Sort(list1):
    n = len(list1)
    for i in range(n):      # Number of passes
        for j in range(0, n-i-1):
        # size -i-1 because last i elements are already sorted
         #in previous passes
            if list1[j] > list1[j+1] :
            # Swap element at jth position with (j+1)th position
                   list1[j], list1[j+1] = list1[j+1], list1[j]
numList = [8, 7, 13, 1, -9, 4]
bubble_Sort(numList)
```

© NCERT not to be republished

Reprint 2026-27

```
print ("The sorted list is :")
for i in range(len(numList)):
    print (numList[i], end=" ")
```

Output:

```
The sorted list is :
-9 1 4 7 8 13
```

Activity 5.2

Apply bubble sort technique to sort a list of elements numList2 = [8, 7, 6, 5, 4]. Show the positions of elements in the list after each pass. In which pass the last swap happens?

5.3 Selection Sort

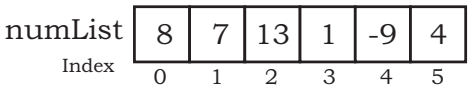
Selection sort is another sorting technique. To sort a list having n elements, the selection sort makes (n-1) number of passes through the list. The list is considered to be divided into two lists -- the left list containing the sorted elements, and the right list containing the unsorted elements. Initially, the left list is empty, and the right list contains all the elements.

For arranging elements in ascending order, in the first pass, all the elements in the unsorted list are traversed to find the smallest element. The smallest element is then swapped with the leftmost element of the unsorted list. This element occupies the first position in the sorted list, and it is not considered in further passes. In the second pass, the next smallest element is selected from the remaining elements in the unsorted list and swapped with the leftmost element of the unsorted list. This element occupies the second position in the sorted list, and the unsorted list reduces by one element for the third pass.

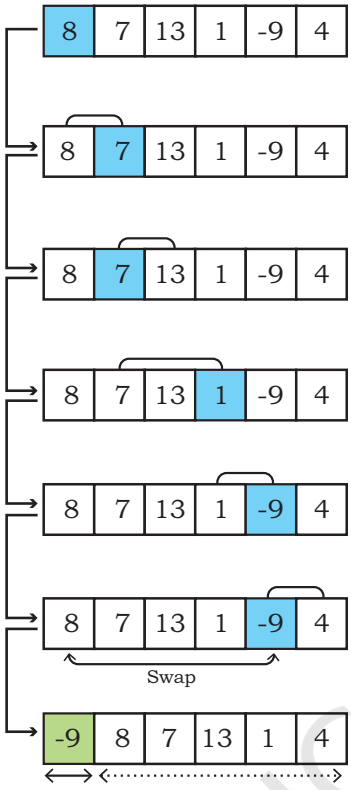
This process continues until n-1 smallest elements are found and moved to their respective places. The nth element is the last, and it is already in place. Figure 5.2 demonstrates the working of selection sort method to arrange a list in ascending order. In this Figure, elements being compared are shown using arrows and the smaller element in a comparison is highlighted with blue colour. The sorted elements are highlighted—

© NCERT not to be republished

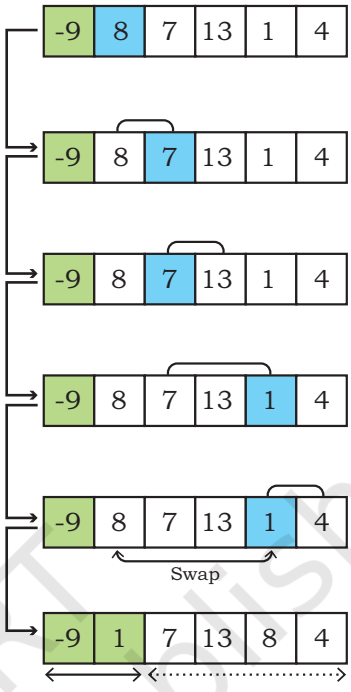
Reprint 2026-27



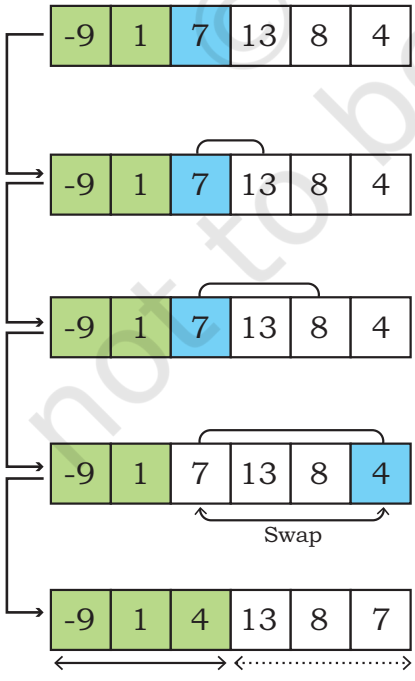
Comparison in Pass 1



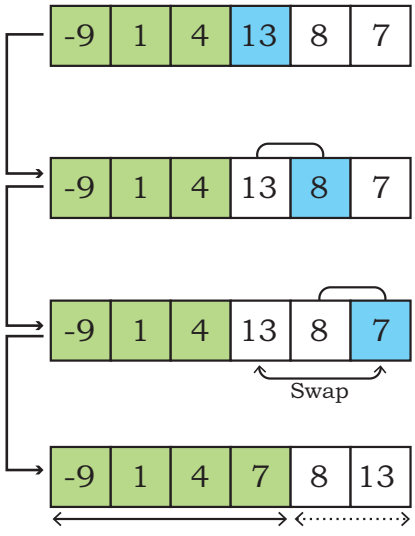
Comparison in Pass 2



Comparison in Pass 3



Comparison in Pass 4



© NCERT not to be republished

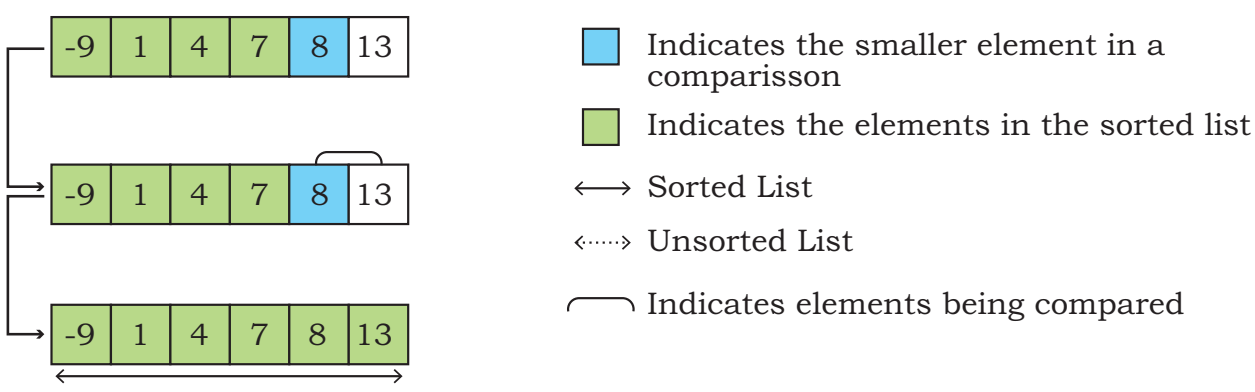


Figure 5.2: Comparisons done in different passes of Selection sort

The following is an algorithm for the selection sort that takes `numList` as a list consisting of n elements, and sorts the list in ascending order:

Algorithm 5.2 shows the steps followed for the selection sort that takes `numList` as a list of n elements, and sorts the list in ascending order:

Algorithm 5.2: Selection Sort

SELECTIONSORT(numList, n)

Step 1: `SET i=0`

Step 2: `WHILE i< n REPEAT STEPS 3 to 11`

Step 3: `SET min = i, flag = 0`

Step 4: `SET j= i+1`

Step 5: `WHILE j<n, REPEAT STEPS 6 to 10`

Step 6: `IF numList[j] < numList[min] THEN`

Step 7: `min = j`

Step 8: `flag = 1`

Step 9: `IF flag = 1 THEN`

Step 10: `swap(numList[i],numList[min])`

Step 11: `SET i=i+1`

Activity 5.3

Consider a list of 10 elements: randList = [7,11,3,10,17,23,1,4,21,5]. Determine the partially sorted list after four complete passes of selection sort.

© NCERT not to be republished

Program 5-2 Implementation of selection sort using Python.

```
def selection_Sort(list2):
   flag = 0            #to decide when to swap
   n=len(list2)
   for i in range(n): # Traverse through all list elements
       min = i
       for j in range(i + 1, len(list2)): #the left elements
       #are already sorted in previous passes
          if list2[j] < list2[min]: # element at j is smaller
          #than the current min element
             min = j
             flag = 1
       if flag == 1 : # next smallest element is found
          list2[min], list2[i] = list2[i], list2[min]

numList = [8, 7, 13, 1, -9, 4]
selection_Sort(numList)
print ("The sorted list is :")
for i in range(len(numList)):
   print (numList[i], end=" ")
```

Output:

```
The sorted list is :
-9 1 4 7 8 13
```

5.4 Insertion Sort

Insertion sort is another sorting algorithm that can arrange elements of a given list in ascending or descending order. Like Selection sort, in Insertion sort also, the list is divided into two parts - one of sorted elements and another of unsorted elements. Each element in the unsorted list is considered one by one and is inserted into the sorted list at its appropriate position. In each pass, the sorted list is traversed from the backward direction to find the position where the unsorted element could be inserted. Hence the sorting method is called insertion sort.

In pass 1, the unsorted list has n-1 elements and the sorted list has a single element (say element s). The first element of the unsorted list (say element e) is compared with the element s of sorted list. If element e is smaller than element s, then element s is shifted to the right making space for inserting element e. This shifting will now make sorted list of size 2 and unsorted list of size n-2.

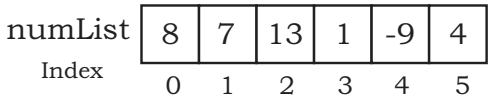
In pass 2, the first element (say element e) of unsorted list will be compared with each element of sorted list

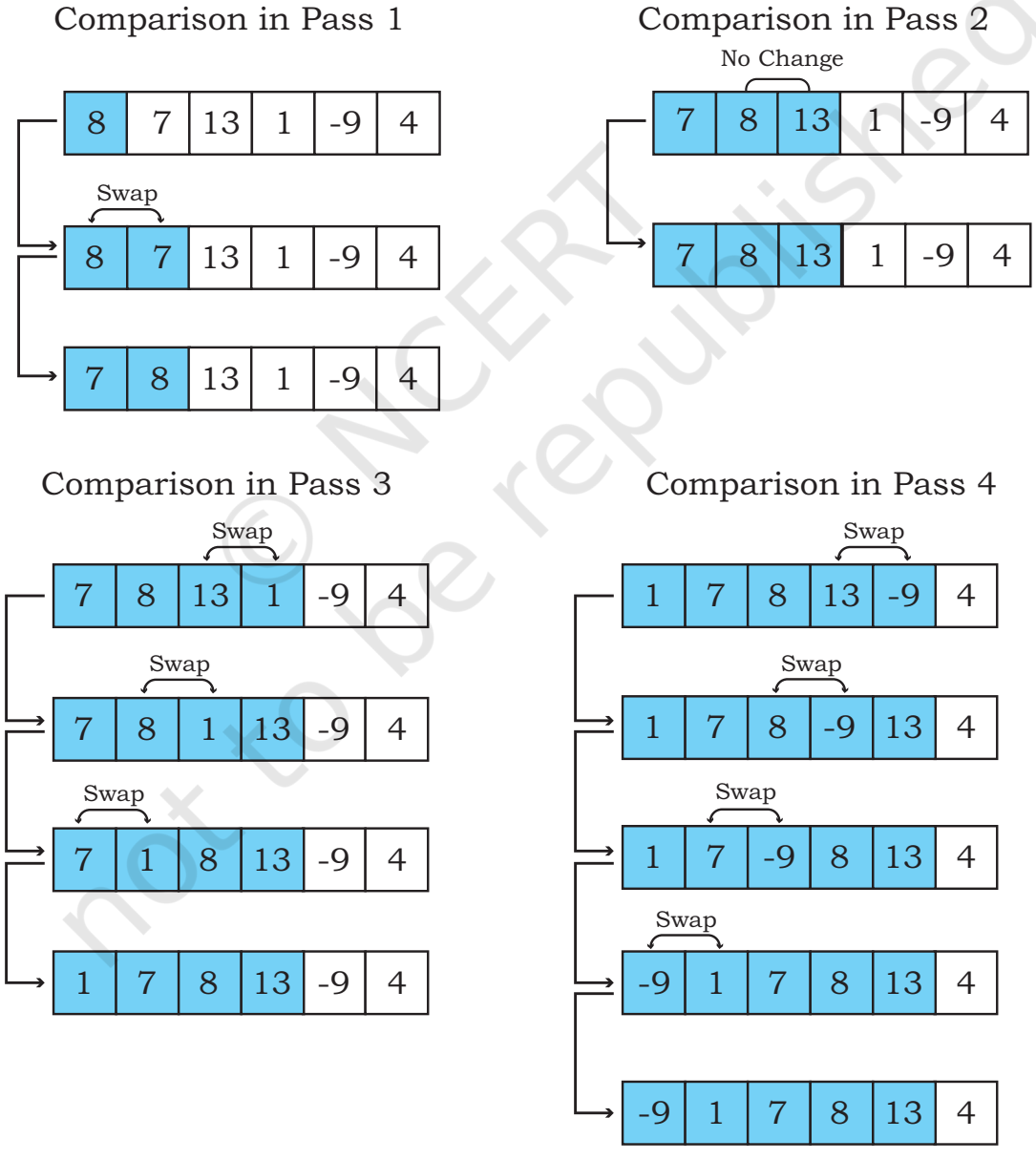
© NCERT not to be republished

starting from the backward direction till the appropriate position for insertion is found. The elements of sorted list will be shifted towards right making space for the element e where it could be inserted.

This continues till all the elements in unsorted lists are inserted at appropriate positions in the sorted list. This results into a sorted list in which elements are arranged in ascending order.

Figure 5.3 demonstrates the working of the insertion sort to arrange a list in ascending order.





© NCERT not to be republished

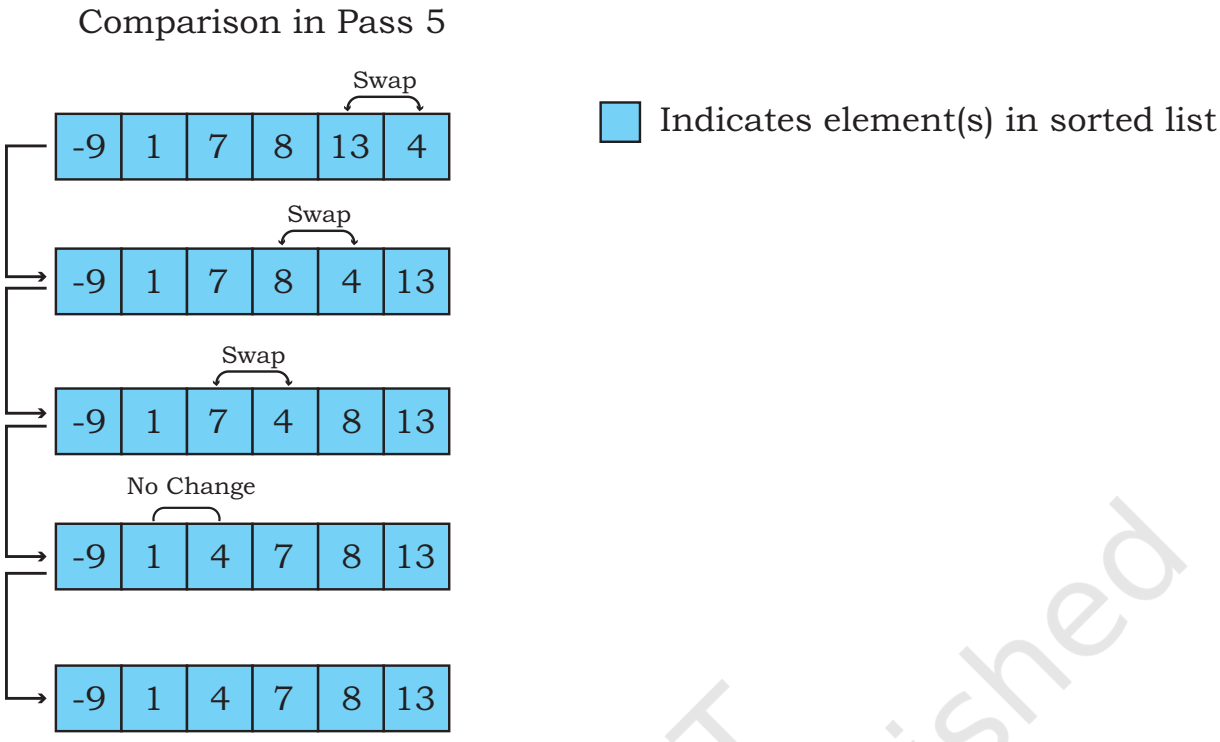


Figure 5.3: Comparisons done in different passes of Insertion sort

Let us consider that numList is a list consisting of n elements. Algorithm 5.3 sorts the list numList in ascending order using insertion sort technique.

Algorithm 5.3: Insertion Sort

INSERTIONSORT(numList, n)

Step 1: SET i=1

Step 2: WHILE i< n REPEAT STEPS 3 to 9

Step 3: temp = numList[i]

Step 4: SET j = i-1

Step 5: WHILE j> = 0 and numList[j]>temp,REPEAT STEPS 6 to 7

Step 6: numList[j+1] = numList[j]

Step 7: SET j=j-1

Step 8: numList[j+1] = temp #insert temp at position j

Step 9: set i=i+1

Activity 5.4

Consider a list of 10 elements:
Array = [7,11,3,10,17,23,1,4,21,5]
Determine the partially sorted list after three complete passes of insertion sort.

© NCERT not to be republished

Program 5-3 Implementation of insertion sort using Python.

```
def insertion_Sort(list3):
     n= len(list3)
     for i in range(n): # Traverse through all elements
          temp = list3[i]
          j = i-1
          while j >=0 and temp< list3[j] :
                 list3[j+1] = list3[j]
                 j = j-1
          list3[j+1] = temp

numList = [8, 7, 13, 1, -9, 4]
insertion_Sort(numList)
print ("The sorted list is :")
for i in range(len(numList)):
    print (numList[i], end=" ")
```

Output:

```
The sorted list is :
-9 1 4 7 8 13
```

5.5 Time Complexity of Algorithms

We have studied that there can be more than one approach to solve a problem using a computer. In Class XI, we compared four different algorithms to check whether a given number is prime or not. For the same problem, one algorithm may require more processing time than the other. The amount of time an algorithm takes to process a given data can be called its time complexity.

For a small set of data elements shown in examples of this chapter so far, the time and memory required by different algorithms do not differ significantly. However, in the real world, sorting algorithms are required to work on huge amounts of data. In such cases, the total time utilisation becomes significant, and therefore it is important to consider the time complexity of an algorithm before being used in a real world data set.

Computer scientists proposing different techniques for sorting are always interested to find out their time complexity. The aim is to find out how a sorting algorithm behaves if the order of input elements changes or if the number of elements in the list increases or decreases. Such comparisons help to decide which algorithm is more suitable for which kind of data and application.

© NCERT not to be republished

Calculating the complexity of different algorithms involves mathematical calculations and detailed analysis, and it is beyond the scope of this textbook to discuss them in detail. However, we will discuss some basics of complexity to get some ideas. The following tips will guide us in estimating the time complexity of an algorithm.

- Any algorithm that does not have any loop will have time complexity as 1 since the number of instructions to be executed will be constant, irrespective of the data size. Such algorithms are known as Constant time algorithms.
- Any algorithm that has a loop (usually 1 to n) will have the time complexity as *n* because the loop will execute the statement inside its body *n* number of times. Such algorithms are known as Linear time algorithms.
- A loop within a loop (nested loop) will have the time complexity as n^2. Such algorithms are known as Quadratic time algorithms.
- If there is a nested loop and also a single loop, the time complexity will be estimated on the basis of the nested loop only.

Now, look at the Python programs of the three sorting techniques discussed in this chapter, you will notice that in each of the three programs, there is a nested loop, i.e., one inside another. So according to the above rules, all the sorting algorithms namely, bubble sort, selection sort and insertion sort have a time complexity of n^2.

SUMMARY

- The process of placing or rearranging a collection of elements into a particular order is known as sorting.
- Bubble sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements in case they are unordered in n-1 passes.
- In Selection Sort, the smallest element is selected from the unsorted array and swapped with the

© NCERT not to be republished

Reprint 2026-27

leftmost element, and that element becomes a part of the sorted array. The process continues for the next element in the unsorted array till the list is sorted.

- Insertion Sort places the element of a list at its suitable place in each pass. It is similar to the placing of cards at its right position while playing cards.
- Complexity analysis is performed to explain how an algorithm will perform when the input grows larger.

EXERCISE

1. Consider a list of 10 elements:

   `numList =[7,11,3,10,17,23,1,4,21,5].`

   Display the partially sorted list after three complete passes of Bubble sort.

2. Identify the number of swaps required for sorting the following list using selection sort and bubble sort and identify which is the better sorting technique with respect to the number of comparisons.

   List 1 :

63	42	21	9

3. Consider the following lists:

   List 1:

2	3	5	7	11

   List 2:

11	7	5	3	2

   If the lists are sorted using Insertion sort then which of the lists List1 or List 2 will make the minimum number of comparisons? Justify using diagrammatic representation.

4. Write a program using user defined functions that accepts a List of numbers as an argument and finds its median. (Hint : Use bubble sort to sort the accepted list. If there are odd number of terms, the median is the center term. If there are even number of terms, add the two middle terms and divide by 2 get median)

© NCERT not to be republished

Reprint 2026-27

5. All the branches of XYZ school conducted an aptitude test for all the students in the age group 14 - 16. There were a total of n students. The marks of n students are stored in a list. Write a program using a user defined function that accepts a list of marks as an argument and calculates the 'xth' percentile (where x is any number between 0 and 100).You are required to perform the following steps to be able to calculate the 'xth' percentile.

   Note: *Percentile is a measure of relative performance i.e.* It is calculated based on a candidate's performance with respect to others. For example : If a candidate's score is in the 90th percentile, that means she/he scored better than 90% of people who took the test.

   Steps to calculate the xth percentile:

   I. Order all the values in the data set from smallest to largest using Selection Sort. In general any of the sorting methods can be used.

   II. Calculate index by multiplying x percent by the total number of values, n.

   For example: to find 90th percentile for 120 students:

   0.90*120 = 108

   III. Ensure that the index is a whole number by using math.round()

   IV. Display the value at the index obtained in Step 3.

   The corresponding value in the list is the xth percentile.

6. During admission in a course, the names of the students are inserted in ascending order. Thus, performing the sorting operation at the time of inserting elements in a list. Identify the type of sorting technique being used and write a program using a user defined function that is invoked every time a name is input and stores the name in ascending order of names in the list.

© NCERT not to be republished

Chapter

6 Searching

12130CH06

"Even though most people won't be directly involved with programming, everyone is affected by computers, so an educated person should have a good understanding of how computer hardware, software, and networks operate."

— Brian Kernighan

In this Chapter

» *Introduction*
» *Linear Search*
» *Binary Search*
» *Search by Hashing*

6.1 Introduction

We store many things in our home and find them out later as and when required. Sometimes we remember the exact location of a required item. But, sometimes we do not remember the exact location and in that case we need to search for the required item. A computer also stores lots of data to be retrieved later as and when demanded by a user or a program.

Searching means locating a particular element in a collection of elements. Search result determines whether that particular element is present in the collection or not. If it is present, we can also find out the position of that element in the given collection. Searching is an important technique in computer science. In order to design algorithms, programmers need to understand the different ways in which a collection of data can be searched for retrieval.

© NCERT not to be republished

6.2 Linear Search

Linear search is the most fundamental and the simplest search method. It is an exhaustive searching technique where every element of a given list is compared with the item to be searched (usually referred to as 'key'). So, each element in the list is compared one by one with the key. This process continues until an element matching the key is found and we declare that the search is successful. If no element matches the key and we have traversed the entire list, we declare the search is unsuccessful i.e., the key is not present in the list. This item by item comparison is done in the order, in which the elements are present in the list, beginning at the first element of the list and moving towards the last. Thus, it is also called sequential search or serial search. This technique is useful for collection of items that are small in size and are unordered.

Given a list `numList` of n elements and key value K, Algorithm 6.1 uses a linear search algorithm to find the position of the key K in `numList`.

Algorithm 6.1 : Linear Search

LinearSearch(numList, key, n)

Step 1: SET index = 0

Step 2: WHILE index < n, REPEAT Step 3

Step 3: IF numlist[index]= key THEN

```
        PRINT "Element found at position", index+1
        STOP
    ELSE
        index = index+1
```

Step 4: PRINT "Search unsuccessful"

Example 6.1 Assume that the numList has seven elements [8, -4, 7, 17, 0, 2, 19] so, n = 7. We need to search for the key, say 17 in numList. Table 6.1 shows the elements in the given list along with their index values.

Table 6.1 Elements in numList alongwith their index value

Index in numList	0	1	2	3	4	5	6
Value	8	-4	7	17	0	2	19

The step-by-step process of linear search using Algorithm 6.1. is given in Table 6.2.

Activity 6.1

Consider a list of 15 elements: L=[2,3,9,7,-6,11,12,17,45,23,29, 31,-37,41,43]. Determine the number of comparisons linear search makes to search for key = 12.

© NCERT not to be republished

Table 6.2 Linear search for key 17 in numList of Table 6.1

index	index < n	numList[index]= key	index=index+1
0	0 < 7 ? Yes	8 = 17? No	1
1	1 < 7 ? Yes	-4 = 17? No	2
2	2 < 7 ? Yes	7 = 17? No	3
3	3 < 7 ? Yes	17 = 17? Yes	

Observe that after four comparisons, the algorithm found the key 17 and will display 'Element found at position 4'.

Let us now assume another arrangement of the elements in numList as [17, 8, -4, 7, 0, 2, 19] and search for the key K=17 in numList.

Table 6.3 Elements in numList alongwith their index value

Index in numList	0	1	2	3	4	5	6
Value	17	8	-4	7	0	2	19

Table 6.4 Linear search for key 17 in numList given in Table 6.3

index	index < n	numList[index]= key	index=index+1
0	0 < 7 ? Yes	17 = 17? Yes	1

From Table 6.4, it is clear that the algorithm had to make only 1 comparison to display 'Element found at position 1'. Thus, if the key to be searched is the first element in the list, the linear search algorithm will always have to make only 1 comparison. This is the minimum amount of work that the linear search algorithm would have to do.

Let us now assume another arrangement of the elements in numList as [8, -4, 7, 0, 2, 19, 17] and search for the key K =17 in numList.

On a dry run, we can find out that the linear search algorithm has to compare each element in the list till the end to display 'Element found at position 7'. Thus, if the key to be searched is the last element in the list, the linear search algorithm will have to make n comparisons, where n is the number of elements in the list. This is in fact the maximum amount of work the linear search algorithm would have to do.

Activity 6.2

In the list : L = [7,-1, 11,32,17,19,23,29,31, 37,43]
Determine the number of comparisons linear search takes to search for key = 43.

© NCERT not to be republished

Let us now assume another case, where the key being searched is not present in the list. For example, we are searching for key = 10 in the numList.

In this case also, the algorithm has to compare each element in the list till the end to display 'Element is not found in the list'. Thus, if the key is not present in the list, the linear search algorithm will have to make n comparisons. This again is a case where maximum work is done. Let us now understand the program of a Linear Search. It takes a list of elements and the key to be searched as input and returns either the position of the element in the list or display that the key is not present in the list.

Program 6-1 Linear Search

```
def linearSearch(list, key):       #function to perform the search
    for index in range(0,len(list)):
        if list[index] == key:     #key is present
            return index+1         #position of key in list
    return None  #key is not in list
#end of function

list1 = []   #Create an empty list
maximum = int(input("How many elements in your list? "))
print("Enter each element and press enter: ")
for i in range(0,maximum):
    n = int(input())
    list1.append(n)     #append elements to the list
print("The List contents are:", list1)

key = int(input("Enter the number to be searched:"))
position = linearSearch(list1, key)
if position is None:
    print("Number",key,"is not present in the list")
else:
    print("Number",key,"is present at position",position)
```

Output

```
How many elements in your list? 4
Enter each element and press enter:
12
23
3
-45
The List contents are: [12, 23, 3, -45]
Enter the number to be searched:23
Number 23 is present at position 2
```

© NCERT not to be republished

6.3 BINARY SEARCH

Consider a scenario where we have to find the meaning of the word Zoology in an English dictionary. Where do we search it in the dictionary?

1. in the first half?
2. around the middle?
3. in the second half?

It is certainly more prudent to look for the word in the second half of the dictionary as the word starts with the alphabet 'Z'. On the other hand, if we were to find the meaning of the word Biology, we would have searched in the first half of the dictionary.

We were able to decide where to search in the dictionary because we are aware of the fact that all words in an English dictionary are placed in alphabetical order. Taking advantage of this, we could avoid unnecessary comparison through each word beginning from the first word of the dictionary and moving towards the end till we found the desired word. However, if the words in the dictionary were not alphabetically arranged, we would have to do linear search to find the meaning of a word.

The binary search is a search technique that makes use of the ordering of elements in the list to quickly search a key. For numeric values, the elements in the list may be arranged either in ascending or descending order of their key values. For textual data, it may be arranged alphabetically starting from a to z or from z to a.

In binary search, the key to be searched is compared with the element in the middle of a sorted list. This could result in either of the three possibilities:

i) the element at the middle position itself matches the key or

ii) the element at the middle position is greater than the key or

iii) the element at the middle position is smaller than the key

If the element at the middle position matches the key, we declare the search successful and the searching process ends.

© NCERT not to be republished

Reprint 2026-27

We are using the term iteration and not comparison in binary search, because after every unsuccessful comparison, we change the search area redefining the first, mid and last position before making subsequent comparisons.

If the middle element is greater than the key it means that if the key is present in the list, it must surely be in the first half. We can thus straight away ignore the second half of the list and repeat the searching process only in the first half.

If the middle element is less than the key, it means if the key is present in the list, it must be in the second half. We can thus straight away ignore the first half of the list and repeat the searching process only in the second half. This splitting and reduction in list size continued till the key is either found or the remaining list consists of only one item. If that item is not the key, then the search is unsuccessful as the key is not in the list.

Thus, it is evident that unlike linear search of elements one-by-one, we can search more efficiently using binary search provided the list from which we want to search is arranged in some order. That is, the list needs to be sorted.

Activity 6.3

Consider the numList [17, 8, -4, 7, 0, 2, 19]. Sort it using the sort() function of Python's Lists. Now apply binary search to search for the key 7. Determine the number of iterations required.

If the list to be searched contains an even number of elements, the mid value is calculated using the floor division (//) operator. If there are 10 elements in the list, then the middle position (mid) = 10//2 = 5. Therefore, the sixth element in the list is considered the middle element as we know that the first element in list has index value 0. If required, the list is further divided into two parts where the first half contains 5 elements and the second half contains 4 elements.

It is interesting to note that the intermediate comparisons which do not find the key still give us information about the part of the list where the key may be found! They reveal whether the key is before or after the current middle position in the list, and we use this information to narrow down or reduce our search area. Thus, each unsuccessful comparison reduces the number of elements remaining to be searched by half, hence the name binary search. Let us now discuss the algorithm of binary search.

Activity 6.4

Consider a list [-4, 0, 2, 7, 8, 17, 19]. Apply binary search to find element -4. Determine the number of key comparisons required.

Given a list numList of n elements and key value K, Algorithm 6.2 shows steps for finding position of the key K in the numList using binary search algorithm.

© NCERT not to be republished

Algorithm 6.2: Binary Search

BinarySearch(numList, key)

Step 1: SET first = 0, last = n-1

Step 2: Calculate mid = (first+last)//2

Step 3: WHILE first <= last REPEAT Step 4

Step 4: IF numList[mid] = key

PRINT "Element found at position", " mid+1

STOP

ELSE

IF numList[mid] > key, THEN last = mid-1

ELSE first = mid + 1

Step 5: PRINT "Search unsuccessful"

The binary search algorithm does not change the list. Rather, after every pass of the algorithm, the search area gets reduced by half. That is, only the index of the element to be compared with the key changes in each iteration.

Example 6.2 Consider a sorted list comprising of 15 elements:

```
numList = [2,3,5,7,10,11,12,17,19,23,29,31,37,41,43]
```

We need to search for the key, say 17 in numList. The first, middle and last element identified in numList alongwith their index values are shown in Table 6.5.

Table 6.5 Elements in sorted numList alongwith their index value

	first				mid									last
Index in numList	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	2	3	5	7	10	11	12	17	19	23	29	31	37	41	43

Table 6.6 Working of binary search using steps given in Algorithm 6.2.

	first	last	mid	$numList_{[mid]}$ == K	key < $L_{mid?}$	first <= last
At Start	0	14	(0+14)//2=7	Not known	Not known	0 <= 14? True
Iteration 1	0	14	7	17 = 17? Yes	Key is found. The search terminates	

Note that the algorithm had to make only 1 iteration to display 'Element found at position 8'. This is because the key being searched is the middle element in the list. Thus, binary search requires only 1 iteration when the key to be searched is the middle value in the list. This

© NCERT not to be republished

is the minimum amount of work binary search would have to do to confirm that a key is present in the list.

Now, let us search for key 2 in the list.

```
numList = [2,3,5,7,10,11,12,17,19,23,29,31,37,41,43]
```

Activity 6.5

For L = [2,3,5,7,10,11,12,17,19,23,29,31,37,41,43]. Fill up the Table 6.8 for the given key values 2, 43, 17 and 9. What do you infer from Table 6.8 regarding performance of both the algorithms in different cases?

In the first iteration, we have the mid value as 17. As 2 is smaller than the mid value (17), we have to search for the first half of the list in the second iteration. We now consider only 7 elements. As 2 is smaller than the new mid value (7), we have to search for the first half of the remaining list in the third iteration. We now consider only 3 elements. Observe that the number of elements in the numList is halved each time. It reduces from 15 elements in iteration 1 to 7 elements in iteration 2, and to 3 elements in iteration 3. In the 3rd iteration, the algorithm finds that key 2 is smaller than the new mid value (3), we have to search in the first half of the remaining list. The list now has only 1 element in the fourth iteration and on comparison, it is found that the element is the same as key. Hence, the search terminates successfully and returns the position of key. Steps followed for binary search are given in Table 6.7.

Table 6.7 Searching key = 2 in the numList using binary search

	first	last	mid	numList[$_{mid}$] == K	K < numList[$_{mid}$]	first <= last
At Start	0	14	(0+14)//2= 7	Not known	Not known	True
Iteration 1	0	14	(0+14)//2= 7	17 = 2? No	2 < 17? True	0 <= 14? True
Iteration 2	0	6	(0+6)//2= 3	7 = 2? No	2 < 7? True	0 <= 6? True
Iteration 3	0	2	(0+2)//2= 1	3 = 2? No	2 < 3? True	0 <= 2? True
Iteration 4	0	0	(0+0)//2= 0	2 = 2? Yes	Key found, search Terminates, return position as (mid+1)=1	

As we can see, the binary search algorithm had to make 4 iterations to narrow down the list to a single element and decide that the search key is the first element of list. This is clearly the maximum work required to find a key in the given list.

© NCERT not to be republished

Program 6-2 Binary Search

```
def binarySearch(list, key):
        first = 0
        last = len(list) - 1
        while(first <= last):
                mid = (first + last)//2
                if list[mid] == key:
                        return mid
                elif key > list[mid]:
                        first = mid + 1
                elif key < list[mid]:
                        last = mid - 1
        return -1

list1 = []   #Create an empty list
print ("Create a list by entering elements in ascending order")
print ("press enter after each element, press -999 to stop")
num = int(input())
while num!=-999:
    list1.append(num)
    num = int(input())
n = int(input("Enter the key to be searched: "))
pos = binarySearch(list1,n)
if(pos != -1):
    print( n,"is found at position", pos+1)
else:
    print (n,"is not found in the list ")
```

Output

```
Create a list by entering elements in ascending order
press enter after each element, press -999 to stop
1
3
4
5
-999
Enter the number to be searched: 4
4 is found at position 3

Second run of the program with different data:
Create a list by entering elements in ascending order
press enter after each element, press -999 to stop
12
8
3
-999
Enter the number to be searched: 4
4 is not found in the list
```

© NCERT not to be republished

Reprint 2026-27

6.3.1 Applications of Binary Search

- Binary search has numerous applications including – searching a dictionary or a telephone directory, finding the element with minimum value or maximum value in a sorted list, etc.
- Modified binary search techniques have far reaching applications such as indexing in databases, implementing routing tables in routers, data compression code, etc.

6.4 Search by Hashing

Hashing is a technique which can be used to know the presence of a key in a list in just one step. The idea is if we already know the value at every index position in a list, it would require only a single comparison to check the presence or absence of a key in that list. Hashing makes searching operations very efficient. A formula called hash function is used to calculate the value at an index in the list.

Think and Reflect

Suppose a list has more than one element whose modulo division results in same remainder value. In such situations, what kind of hashing may be useful?

Thus, a *hash function* takes elements of a list one by one and generates an index value for every element. This will generate a new list called the hash table. Each index of the hash table can hold only one item and the positions are indexed by integer values starting from 0. Note that the size of the hash table can be larger than the size of the list.

A simple hash function that works with numeric values is known as the *remainder method*. It takes an element from a list and divides it by the size of the hash table. The remainder so generated is called the hash value.

h(element) = element % size(hash table)

We can easily implement a hash table using a Python's List. Let us consider an empty hash table having 10 positions as shown in Table 6.8:

Table 6.8 An Empty hash table with 10 positions

Index/ position	0	1	2	3	4	5	6	7	8	9
Value	None	None	None	None	None	None	None	None	None	None

Let us consider a list of numbers (34, 16, 2, 93, 80, 77, 51). We can use the hash function remainder

© NCERT not to be republished

Reprint 2026-27

method explained earlier to create a hash table as shown in Table 6.9.

Table 6.9 hash function element % 10 applied on the elements of list

Element	34	16	2	93	80	77	51
Hash Value	34%10=4	16%10=6	2%10=2	93%10=3	80%10=0	77%10=7	51%10=1

After computing the hash values, each element is inserted at its designated position in the hash table shown in Table 6.10

Table 6.10 hash table generated for elements given in Table 6.10

index	0	1	2	3	4	5	6	7	8	9
Value	80	51	2	93	34	None	16	77	None	None

Now, to search for a key, we can calculate its index using the hashing function and compare the element at that index with the key to declare whether the element is present in the list or not. This search operation involves just one comparison and hence the same amount of time is always required to search a key irrespective of the size of the list.

Program 6-3 Use of hashing to find a key in the given list L

```
#Function to check if a key is present or not
def hashFind(key,hashTable):
    if (hashTable[key % 10] == key): #key is present
        return ((key % 10)+1)   #return the position
    else:
        return None          #key is not present
#end of function

#create hashTable with 10 empty positions
hashTable=[None, None, None, None, None, None, None, None, None,
None]
print("We have created a hashTable of 10 positions:")
print(hashTable)

L = [34, 16, 2, 93, 80, 77, 51]
print("The given list is", L[::] )

# Apply hash function
for i in range(0,len(L)):
    hashTable[L[i]%10] = L[i]
```

© NCERT not to be republished

```
print("The hash table contents are: " )
for i in range(0,len(hashTable)):
    print("hashindex=", i," , value =", hashTable[i])

key = int(input("Enter the number to be searched:"))

position = hashFind(key,hashTable)
if position is None:
    print("Number",key,"is not present in the hash table")
else:
    print("Number ",key," present at ",position, " position")
```

Output:

```
We have created a hashTable of 10 positions:
[None, None, None, None, None, None, None, None, None, None]
The given list is [34, 16, 2, 93, 80, 77, 51]
The hash table contents are:
hashindex= 0  , value = 80
hashindex= 1  , value = 51
hashindex= 2  , value = 2
hashindex= 3  , value = 93
hashindex= 4  , value = 34
hashindex= 5  , value = None
hashindex= 6  , value = 16
hashindex= 7  , value = 77
hashindex= 8  , value = None
hashindex= 9  , value = None
Enter the number to be searched:16
Number  16  present at  7  position
```

6.4.1 COLLISION

The hashing technique works fine if each element of the list maps to a unique location in the hash table. Consider a list `[34, 16, 2, 26, 80]`. While applying the hash function say, list `[i]%10`, two elements (16 and 26) would have a hash value 6. This is a problematic situation, because according to our definition, two or more elements cannot be in the same position in the list. This situation is called *collision* in hashing.

We must have a mechanism for placing the other items with the same hash value in the hash table. This process is called *collision resolution*. Collision can be resolved in many ways, but it is beyond the scope of this book to discuss collision resolution methods.

© NCERT not to be republished

If every item of the list maps to a unique index in the hash table, the hash function is called a *perfect hash function*. If a hash function is perfect, collision will never occur.

Apart from modulo division method, hash functions may be based on several other techniques like integer division, shift folding, boundary folding, mid-square function, extraction, radix transformation, etc. Again, it is beyond the scope of this book to discuss these methods.

The time taken by different hash functions may be different, but it remains constant for a particular hash function. The advantage of hashing is that the time required to compute the index value is independent of the number of items in the search list. It is to remember that the cost of computing a hash function must be small enough to make a hashing-based searching more efficient than other search methods.

SUMMARY

- Searching means trying to locate a particular element called key in a collection of elements. Search specifies whether that key is present in the collection or not. Also, if the key is present, it tells the position of that key in the given collection.
- Linear search checks the elements of a list, one at a time, without skipping any element. It is useful when we need to search for an item in a small unsorted list, but it is slow and time-consuming when the list contains a large number of items. The time taken to search the list increases as the size of the list increases.
- Binary search takes a sorted/ordered list and divides it in the middle. It then compares the middle element with the key to be searched. If the middle element matches the key, the search is declared successful and the program ends. If the middle element is greater than the key, the search repeats only in the first half of the list. If the middle element is lesser than the key, the search repeats only in the second half of the list.

© NCERT not to be republished

Reprint 2026-27

This splitting and reduction in list size continue till the key is found or the remaining list consists of only one item.

- In binary search, comparisons that do not find the key still give us idea about the location where the key may probably be found! They reveal whether the key is before or after the current middle position in the list, and we can use this information to narrow down or reduce our searching efforts.
- Hash based searching requires only one key comparison to discover the presence or absence of a key, provided every element is present at its designated position decided by a hash function. It calculates the position of the key in the list using a formula called the hash function and the key itself.
- When two elements map to the same slot in the hash table, it is called collision.
- The process of identifying a slot for the second and further items in the hash table in the event of collision, is called collision resolution.
- A perfect hash function maps every input key to a unique index in the hash table. If the hash function is perfect, collisions will never occur.

EXERCISE

1. Using linear search determine the position of 8, 1, 99 and 44 in the list:

```
[1, -2, 32, 8, 17, 19, 42, 13, 0, 44]
```

   Draw a detailed table showing the values of the variables and the decisions taken in each pass of linear search.

2. Use the linear search program to search the key with value 8 in the list having duplicate values such as `[42, -2, 32, 8, 17, 19, 42, 13, 8, 44]`. What is the position returned? What does this mean?

3. Write a program that takes as input a list having a mix of 10 negative and positive numbers and a key value.

© NCERT not to be republished

Reprint 2026-27

Apply linear search to find whether the key is present in the list or not. If the key is present it should display the position of the key in the list otherwise it should print an appropriate message. Run the program for at least 3 different keys and note the result.

4. Write a program that takes as input a list of 10 integers and a key value and applies binary search to find whether the key is present in the list or not. If the key is present it should display the position of the key in the list otherwise it should print an appropriate message. Run the program for at least 3 different key values and note the results.

5. Following is a list of unsorted/unordered numbers:

   `[50, 31, 21, 28, 72, 41, 73, 93, 68, 43, 45, 78, 5, 17, 97, 71, 69, 61, 88, 75, 99, 44, 55,9]`

   - Use linear search to determine the position of 1, 5, 55 and 99 in the list. Also note the number of key comparisons required to find each of these numbers in the list.
   - Use a Python function to sort/arrange the list in ascending order.
   - Again, use linear search to determine the position of 1, 5, 55 and 99 in the list and note the number of key comparisons required to find these numbers in the list.
   - Use binary search to determine the position of 1, 5, 55 and 99 in the sorted list. Record the number of iterations required in each case.

6. Write a program that takes as input the following unsorted list of English words:

   `[Perfect, Stupendous, Wondrous, Gorgeous, Awesome, Mirthful, Fabulous, Splendid, Incredible, Outstanding, Propitious, Remarkable, Stellar, Unbelievable, Super, Amazing].`

   - Use linear search to find the position of `Amazing, Perfect, Great` and `Wondrous` in the list. Also note the number of key comparisons required to find these words in the list.
   - Use a Python function to sort the list.
   - Again, use linear search to determine the position of `Amazing, Perfect, Great` and `Wondrous` in the list and note the number of key comparisons required to find these words in the list.
   - Use binary search to determine the position of `Amazing, Perfect, Great` and `Wondrous` in the sorted list. Record the number of iterations required in each case.

© NCERT not to be republished

Reprint 2026-27

7. Estimate the number of key comparisons required in binary search and linear search if we need to find the details of a person in a sorted database having 230 (1,073,741,824) records when details of the person being searched lies at the middle position in the database. What do you interpret from your findings?

8. Use the hash function: `h(element)= element%11` to store the collection of numbers: `[44, 121, 55, 33, 110, 77, 22, 66]` in a hash table. Display the hash table created. Search if the values 11, 44, 88 and 121 are present in the hash table, and display the search results.

9. Write a Python program by considering a mapping of list of countries and their capital cities such as:

```
CountryCapital= {'India':'New Delhi','UK':
                 'London','France':'Paris',
                 'Switzerland': 'Berne',
                 'Australia': 'Canberra'}
```

Let us presume that our hash function is the length of the Country Name. Take two lists of appropriate size: one for keys (Country) and one for values (Capital). To put an element in the hash table, compute its hash code by counting the number of characters in Country, then put the key and value in both the lists at the corresponding indices. For example, India has a hash code of 5. So, we store India at the 5th position (index 4) in the keys list, and New Delhi at the 5th position (index 4) in the values list and so on. So that we end up with:

hash index = length of key - 1	List of Keys	List of Values
0	None	None
1	UK	London
2	None	None
3	Cuba	Havana
4	India	New Delhi
5	France	Paris
6	None	None
7	None	None
8	Australia	Canberra
9	None	None
10	Switzerland	Berne

Now search the capital of India, France and the USA in the hash table and display your result.

© NCERT not to be republished

Chapter 7

Understanding Data

12130CH07

"Data is not information, Information is not knowledge, Knowledge is not understanding, Understanding is not wisdom."

— Gary Schubert

In this Chapter

- *Introduction to Data*
- *Data Collection*
- *Data Storage*
- *Data Processing*
- *Statistical Techniques for Data Processing*

7.1 Introduction to Data

Many a time, people take decisions based on certain data or information. For example, while choosing a college for getting admission, one looks at placement data of previous years of that college, educational qualification and experience of the faculty members, laboratory and hostel facilities, fees, etc. So we can say that identification of a college is based on various data and their analysis. Governments systematically collect and record data about the population through a process called census. Census data contains valuable information which are helpful is planning and formulating policies. Likewise, the coaching staff of a sports team analyses previous performances of opponent teams for making strategies. Banks maintain data about the customers, their account details and transactions. All these examples highlight the need of data in various fields. Data are indeed crucial for decision making.

© NCERT not to be republished

In the previous examples, one cannot make decisions by looking at the data itself. In our example of choosing a college, suppose the placement cell of the college has maintained data of about 2000 students placed with different companies at different salary packages in the last 3 years. Looking at such data, one cannot make any remark about the placement of students of that college. The college processes and analyses this data and the results are given in the placement brochure of the college through summarisation as well as visuals for easy understanding. Hence, data need to be gathered, processed and analysed for making decisions.

A knowledge base is a store of information consisting of facts, assumptions and rules which an AI system can use for decision making.

In general, data is a collection of characters, numbers, and other symbols that represents values of some situations or variables. Data is plural and singular of the word data is "datum". Using computers, data are stored in electronic forms because data processing becomes faster and easier as compared to manual data processing done by people. The Information and Communication Technology (ICT) revolution led by computer, mobile and Internet has resulted in generation of large volume of data and at a very fast pace. The following list contains some examples of data that we often come across.

- Name, age, gender, contact details, etc., of a person
- Transactions data generated through banking, ticketing, shopping, etc. whether online or offline
- Images, graphics, animations, audio, video
- Documents and web pages
- Online posts, comments and messages
- Signals generated by sensors
- Satellite data including meteorological data, communication data, earth observation data, etc.

7.1.1 Importance of Data

Human beings rely on data for making decisions. Besides, large amount of data when processed with the help of a computer, show us the possibilities or hidden traits which are otherwise not visible to humans. When one withdraws money from ATM, the bank needs to debit the withdrawn amount from the linked account. So the bank needs to maintain data and update it as and when required. The meteorological offices continuously keep on monitoring satellite data for any upcoming cyclone or heavy rain.

© NCERT not to be republished

In a competitive business environment, it is important for business organisations to continuously monitor and analyse market behaviour with respect to their products and take actions accordingly. Besides, companies identify customer demands as well as feedbacks, and make changes in their products or services accordingly.

The dynamic pricing concept used by airlines and railway is another example where they decide the price based on relationships between demand and supply. The cab booking Apps increase or decrease the price based on demand for cabs at a particular time. Certain restaurants offer discounted price (called happy hours), they decide when and how much discount to offer by analysing sales data at different time periods.

Besides business, following are some other scenarios where data are also stored and analysed for making decisions:

- The electronic voting machines are used for recording the votes cast. Subsequently, the voting data from all the machines are accumulated to declare election results in a short time as compared to manual counting of ballot papers.
- Scientists record data while doing experiments to calculate and compare results.
- Pharmaceutical companies record data while trying out a new medicine to see its effectiveness.
- Libraries maintain data about books in the library and the membership of the library.
- The search engines give us results after analysing large volume of data available on the websites across World Wide Web (www).
- Weather alerts are generated by analysing data received from various satellites.

7.1.2 Types of Data

As data come from different sources, they can be in different formats. For example, an image is a collection of pixels; a video is made up of frames; a fee slip is made up of few numeric and non-numeric entries; and messages/chats are made up of texts, icons (emoticons) and images/videos. Two broad categories in which data can be classified on the basis of their format are:

(A) Structured Data

Data which is organised and can be recorded in a well defined format is called structured data. Structured

© NCERT not to be republished

Reprint 2026-27

Activity 7.1

Observe Voter Identity cards of your family members and identify the data fields under which data are organised. Are they same for all?

data is usually stored in computer in a tabular (in rows and columns) format where each column represents different data for a particular parameter called attribute/ characteristic/variable and each row represents data of an observation for different attributes. Table 7.1 shows structured data related to an inventory of kitchen items maintained by a shop.

Table 7.1 Structured data about kitchen items in a shop

ModelNo	ProductName	Unit Price	Discount(%)	Items_in_Inventory
ABC1	Water bottle	126	8	13
ABC2	Melamine Plates	320	5	45
ABC3	Dinner Set	4200	10	8
GH67	Jug	80	0	10
GH78	Table Spoon	120	5	14
GH81	Bucket	190	12	6
NK2	Kitchen Towel	25	0	32

Given this data, using a spreadsheet or other such software, the shop owner can find out how many total items are there by summing the column Items_in_ Inventory of Table 7.1 The owner of the shop can also calculate the total value of all items in the inventory by multiplying each entry of column 3 (Unit Price) with corresponding entry of column 5 (Items_in_Inventory) and finding their sum.

Table 7.2 shows more examples of structured data recorded for different attributes.

Table 7.2 Attributes maintained for different activities

Entity/Activities	Data Fields/Parameters/Attributes
Books at a shop	BookTitle, Author, Price, YearofPublication
Depositing fees in a school	StudentName, Class, RollNo, FeesAmount, DepositDate
Amount withdrawal from ATM	AccHolderName, AccountNo, TypeofAcc, DateofWithdrawal, AmountWithdrawn, ATMid, TimeOfWithdrawal

(B) Unstructured Data

A newspaper contains various types of news items which are also called data. But there is no fixed pattern that a newspaper follows in placing news articles. One day there might be three images of different sizes on a page along with five news items and one or more advertisements. While on another day there, might be one big image with three textual news items. So there is

© NCERT not to be republished

no particular format nor any fixed structure for printing news. Another example is the content of an email. There is no fixed structure about how many lines or paragraphs one has to write in an email or how many files are to be attached with an email. In summary, data which are not in the traditional row and column structure is called unstructured data.

Examples of unstructured data include web pages consisting of text as well as multimedia contents (image, graphics, audio/video). Other examples include text documents, business reports, books, audio/video files, social media messages. Although there are ways to process unstructured data, we are going to focus on handling structured data only in this book.

Unstructured data are sometimes described with the help of some other data called metadata. Metadata is basically data about data. For example, we describe different parts of an email as subject, recipient, main body, attachment, etc. These are the metadata for the email data. Likewise, we can have some metadata for an image file as image size (in KB or MB), image type (for example, JPEG, PNG), image resolution, etc.

Think and Reflect

When we click a photograph using our digital or mobile camera, does it have some metadata associated with it?

7.2 Data Collection

For processing data, we need to collect or gather data first. We can then store the data in a file or database for later use. Data collection here means identifying already available data or collecting from the appropriate sources. Suppose there are three different scenarios where sales data in a grocery store are available:

- Sales data are available with the shopkeeper in a diary or register. In this case we should enter the data in a digital format for example, in a spreadsheet.
- Data are already available in a digital format, say in a CSV (comma separated values) file.
- The shopkeeper has so far not recorded any data in either form but wants to get a software developed for maintaining sales data and accounts. The software may be developed using a programming language such as Python which can be used to store and retrieve data from a CSV file or a database management system like MySQL, which will be discussed further.

© NCERT not to be republished

Reprint 2026-27

Data are continuously being generated at different sources. Our interactions with digital medium are continuously generating huge volumes of data. Hospitals are collecting data about patients for improving their services. Shopping malls are collecting data about the items being purchased by people. On analysing such data, suppose it appears that bedsheets and groceries are frequently bought together. Hence, the shop owner may decide to display bedsheets near the grocery section in the mall to increase the sales. Likewise, a political analyst may look at the data contained in the posts and messages at a social media platform and analyse to see public opinion before an election. Organisations like World Bank and International Monetary Fund (IMF) are collecting data related to various economic parameters from different countries for making economic forecasts.

Think and Reflect

Identify attributes needed for creating an Aadhaar Card.

7.3 Data Storage

Once we gather data and process them to get results, we may not then simply discard the data. Rather, we would like to store them for future use as well. Data storage is the process of storing data on storage devices so that data can be retrieved later. Now a days large volume of data are being generated at a very high rate. As a result, data storage has become a challenging task. However, the decrease in the cost of digital storage devices has helped in simplifying this task. There are numerous digital storage devices available in the market like, Hard Disk Drive (HDD), Solid State Drive (SSD), CD/DVD, Tape Drive, Pen Drive, Memory Card, etc.

Think and Reflect

Is it necessary to store data in files before processing?

We store data like images, documents, audios/ videos, etc. as files in our computers. Likewise, school/ hospital data are stored in data files. We use computers to add, modify or delete data in these files or process these data files to get results. However, file processing has certain limitations, which can be overcome through Database Management System (DBMS).

7.4 Data Processing

We are interested in understanding data as they hold valuable facts and information that can be useful in our decision making process. However, by looking at the vast or large amount of data, one cannot arrive at a conclusion. Rather, data need to be processed to

© NCERT not to be republished

get results and after analysing those results, we make conclusions or decisions.

We find automated data processing in situations like online bill payment, registration of complaints, booking tickets, etc. Figure 7.1 illustrates basic steps used to process the data to get the output.

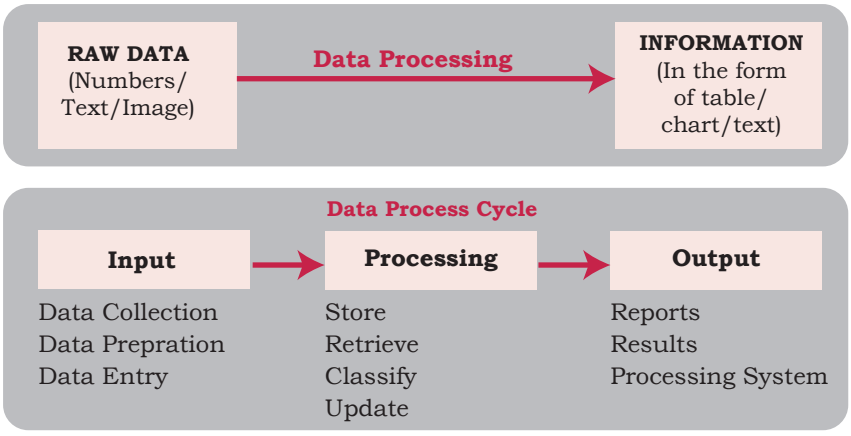


Figure 7.1: Steps in data processing

Figure 7.2 shows some tasks along with data, processing and generated output/information.

7.5 Statistical Techniques for Data Processing

Given a set of data values, we need to process them to get information. There are various techniques which help us to have preliminary understanding about the data.

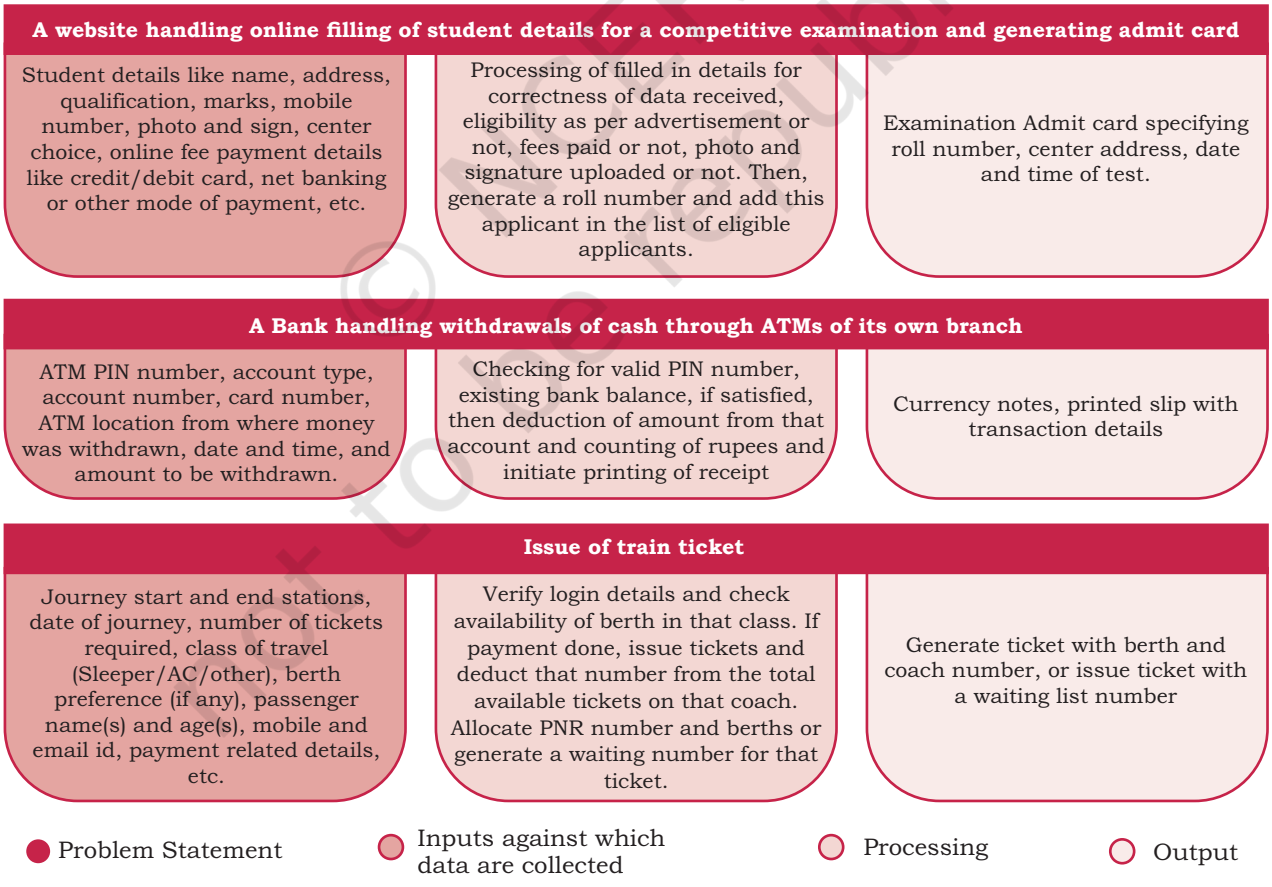


Figure 7.2: Data based problem statements

© NCERT not to be republished

Reprint 2026-27

Summarisation methods are applied on tabular data for its easy comprehension. Commonly used statistical techniques for data summarisation are given below:

7.5.1 Measures of Central Tendency

A measure of central tendency is a single value that gives us some idea about the data. Three most common measures of central tendency are the *mean*, *median*, and *mode*. Instead of looking at each individual data values, we can calculate the mean, median and mode of the data to get an idea about average, middle value and frequency of occurrence of a particular value, respectively. Selection of a measure of central tendency depends on certain characteristics of data.

(A) Mean

Mean is simply the average of numeric values of an attribute. Mean is also called *average*. Suppose there are data on weight of 40 students in a class. Instead of looking at each of the data values, we can calculate the average to get an idea about the average weight of students in that class.

Definition: Given n values x_1, x_2, x_3,...x_n, mean is computed as $\frac{\sum_{i}^{n} x_i}{n}$.

Example 7.1

Assume that height (in cm) of students in a class are as follows [90,102,110,115,85,90,100,110,110]. Mean or average height of the class is

$$\frac{90+102+110+115+85+90+100+110+110}{9} = \frac{912}{9} = 101.33\,\text{cm}$$

Mean is not a suitable choice if there are outliers in the data. To calculate mean, the outliers or extreme values should be removed from the given data and then calculate mean of the remaining data.

Note: An outlier is an exceptionally large or small value, in comparison to other values of the data. Usually, outliers are considered as error since they can influence/affect the average or other statistical calculation based on the data.

(B) Median

Median is also computed for a single attribute/variable at a time. When all the values are sorted in ascending or descending order, the middle value is called the *Median*. When there are odd number of values, then median is

© NCERT not to be republished

Reprint 2026-27

the value at the middle position. If the list has even number of values, then median is the average of the two middle values. Median represents the central value at which the given data is equally divided into two parts.

Example 7.2

Consider the previous data of height of students used in calculation of mean value. In order to compute the median, the first step is to sort data in ascending or descending order. We have sorted the height data in ascending order as [85,90,90,100,102,110,110,110, 115]. As there are total 9 values (odd number), the median is the value at position 5, that is 102 cm, whether counted from left to right or from right to left. Median represents the actual central value at which the given data is equally divided into two parts.

Think and Reflect

Out of Mean and Median, which one is more sensitive to outliers in data?

(C) Mode

Value that appears most number of times in the given data of an attribute/variable is called *Mode*. It is computed on the basis of frequency of occurrence of distinct values in the given data. A data set has no mode if each value occurs only once. There may be multiple modes in the data if more than one values have same highest frequency. Mode can be found for numeric as well as non-numeric data.

Example 7.3

In the list of height of students, mode is 110 as its frequency of occurrence in the list is 3, which is larger than the frequency of rest of the values.

7.5.2 Measures of Variability

The measures of variability refer to the spread or variation of the values around the mean. They are also called measures of dispersion that indicate the degree of diversity in a data set. They also indicate difference within the group. Two different data sets can have the same mean, median or mode but completely different levels of dispersion, or vice versa. Common measures of dispersion or variability are Range and Standard Deviation.

(A) Range

It is the difference between maximum and minimum values of the data (the largest value minus the smallest value). *Range* can be calculated only for numerical data. It is a measure of dispersion and tells about coverage/spread of data values. For

© NCERT not to be republished

Reprint 2026-27

example difference in salaries of employees, marks of a student, price of toys, etc. As range is calculated based on the two extreme values, any outlier in the data badly influences the result.

Let *M* be the largest or maximum value and *S* is the smallest or minimum value in the data, then Range is the difference between two extreme values i.e. *M – S or Maximum – Minimum.*

Example 7.4

In the above example, minimum hight value is 85 cm and maximum hight value is 115 cm. Hence, range is 115-85 = 30 cm.

(B) Standard deviation

Standard deviation refers to differences within the group or set of data of a variable. Like Range, it also measures the spread of data. However, unlike Range which only uses two extreme values in the data, calculation of standard deviation considers all the given data. It is calculated as the positive square root of the average of squared difference of each value from the mean value of data. Smaller value of standard deviation means data are less spread while a larger value of standard deviation means data are more spread.

Given n values x_1, x_2, x_3,...x_n, and their mean $\bar{x}$, the standard deviation, represented as σ (greek letter sigma) is computed as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n}}$$

Example 7.5

Let us compute the standard deviation of the height of nine students that we used while calculating Mean. The Mean ($\bar{x}$) was calculated to be 101.33 cm. Subtract each value from the mean and take square of that value. Dividing the sum of square values by total number of values and taking its square not gives the standard deviation in data. See Table 7.3 for details.

Table 7.3 Standard deviation of attendance of 9 students

Height (x) in cm	$x - \bar{x}$	$(x - \bar{x})^2$	$= \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n}$
90	-11.33	128.37	
102	0.67	0.36	
110	8.67	75.17	
115	13.67	186.87	

© NCERT not to be republished

85	-16.33	266.67	$= \frac{938}{9} = 104.22$
90	-11.33	128.37	
100	-1.33	1.77	$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n}}$
110	8.67	75.17	
110	8.67	75.17	
$n = 9$ $\bar{x} = 101.33$	$\sum(x-\bar{x}) = 0.03$	$\sum(x-\bar{x})^2 = 938.00$	$= \sqrt{104.22} = 10.2\,\text{cm}$

Let us look at the following problems and select the suitable statistical technique to be applied (Mean/Median/Mode/Range/Standard Deviation):

Problem Statement	Choose suitable statistical method
The management of a company wants to know about disparity in salaries of all employees.	
Teacher wants to know about the average performance of the whole class in a test.	
Compare height of residents of two cities	
Find the dominant value from a set of values	
Compare income of residents of two cities	
Find the popular color for car after surveying the car owners of a small city.	

It is important to understand statistical techniques so that one can decide which statistical technique to use to arrive at a decision. Different programming tools are available for efficient analysis of large volumes of data. These tools make use of statistical techniques for data analysis. One such programming tool is Python and it has libraries specially built for data processing and analysis. We will be covering some of them in the following chapters.

Summary

- Data refer to unorganised facts that can be processed to generate meaningful result or information.
- Data can be structured or unstructured.
- Hard Disk, SSD, CD/DVD, Pen Drive, Memory Card, etc. are some of the commonly used storage devices.

© NCERT not to be republished

Reprint 2026-27

- Data Processing cycle involves input and storage of data, its processing and generating output.
- Summarising data using statistical techniques aids in revealing data characteristics.
- Mean, Median, Mode, Range, and Standard Deviation are some of the statistical techniques used for data summarisation.
- Mean is the average of given values.
- Median is the mid value when data are sorted in ascending/descending order.
- Mode is the data value that appears most number of times.
- Range is the difference between the maximum and minimum values.
- Standard deviation is the positive square root of the average of squared difference of each value from the mean.

EXERCISE

1. Identify data required to be maintained to perform the following services:
   a) Declare exam results and print e-certificates
   b) Register participants in an exhibition and issue biometric ID cards
   c) To search for an image by a search engine
   d) To book an OPD appointment with a hospital in a specific department
2. A school having 500 students wants to identify beneficiaries of the merit-cum means scholarship, achieving more than 75% for two consecutive years and having family income less than 5 lakh per annum. Briefly describe data processing steps to be taken by the to beneficial prepare the list of school.
3. A bank 'xyz' wants to know about its popularity among the residents of a city 'ABC' on the basis of number of bank accounts each family has and the average monthly account balance of each person. Briefly describe the steps to be taken for collecting data and what results can be checked through processing of the collected data.

© NCERT not to be republished

Reprint 2026-27

4. Identify type of data being collected/generated in the following scenarios:
   a) Recording a video
   b) Marking attendance by teacher
   c) Writing tweets
   d) Filling an application form online

5. Consider the temperature (in Celsius) of 7 days of a week as 34, 34, 27, 28, 27, 34, 34. Identify the appropriate statistical technique to be used to calculate the following:
   a) Find the average temperature.
   b) Find the temperature Range of that week.
   c) Find the standard deviation temperature.

6. A school teacher wants to analyse results. Identify the appropriate statistical technique to be used along with its justification for the following cases:
   a) Teacher wants to compare performance in terms of division secured by students in Class XII A and Class XII B where each class strength is same.
   b) Teacher has conducted five unit tests for that class in months July to November and wants to compare the class performance in these five months.

7. Suppose annual day of your school is to be celebrated. The school has decided to felicitate those parents of the students studying in classes XI and XII, who are the alumni of the same school. In this context, answer the following questions:
   a) Which statistical technique should be used to find out the number of students whose both parents are alumni of this school?
   b) How varied are the age of parents of the students of that school?

8. For the annual day celebrations, the teacher is looking for an anchor in a class of 42 students. The teacher would make selection of an anchor on the basis of singing skill, writing skill, as well as monitoring skill.
   a) Which mode of data collection should be used?
   b) How would you represent the skill of students as data?

9. Differentiate between structured and unstructured data giving one example.

   The principal of a school wants to do following analysis on the basis of food items procured and sold in the canteen:
   a) Compare the purchase and sale price of fruit juice and biscuits.

© NCERT not to be republished

Reprint 2026-27

b) Compare sales of fruit juice, biscuits and samosa.
c) Variation in sale price of fruit juices of different companies for same quantity (in ml).

Create an appropriate dataset for these items (fruit juice, biscuits, samosa) by listing their purchase price and sale price. Apply basic statistical techniques to make the comparisons.

© NCERT not to be republished

Chapter 8

Database Concepts

12130CH08

"Inconsistency of your mind... Can damage your memory... Remove the inconsistent data... And keep the original one !!!"

— Nisarga Jain

In this Chapter

- *Introduction*
- *File System*
- *Database Management System*
- *Rational Data Model*
- *Keys in a Relational Database*

8.1 Introduction

After learning about importance of data in the previous chapter, we need to explore the methods to store and manage data electronically. Let us take an example of a school that maintains data about its students, along with their attendance record and guardian details.

The class teacher marks daily attendance of the students in the attendance register. The teacher records 'P' for present or 'A' for absent against each student's roll number on each working day. If class strength is 50 and total working days in a month are 26, the teacher needs to record 50×26 records manually in the register every month. As the volume of data increases, manual data entry becomes tedious. Following are some of the limitations of manual record keeping in this example:

© NCERT not to be republished

Activity 8.1

Visit a few shops where records are maintained manually and identify a few limitations of manual record keeping faced by them.

1) Entry of student details (Roll number and name) in the new attendance register when the student is promoted to the next class.
2) Writing student details on each month's attendance page where inconsistency may happen due to incorrectly written names, skipped student records, etc.
3) Loss of data in case attendance register is lost or damaged.
4) Erroneous calculation while consolidating attendance record manually.

The office staff also manually maintain student details viz. Roll Number, Name and Date of Birth with respective guardian details viz. Guardian name, Contact Number and Address. This is required for correspondence with guardian regarding student attendance and result.

Finding information from a huge volume of papers or deleting/modifying an entry is a difficult task in pen and paper based approach. To overcome the hassles faced in manual record keeping, it is desirable to store attendance record and student details on separate data files on a computerised system, so that office staff and teachers can:

1) Simply copy the student details to the new attendance file from the old attendance file when students are promoted to next class.
2) Find any data about student or guardian.
3) Add more details to existing data whenever a new student joins the school.
4) Modify stored data like details of student or guardian whenever required.
5) Remove/delete data whenever a student leaves the school.

8.2 File System

A file can be understood as a container to store data in a computer. Files can be stored on the storage device of a computer system. Contents of a file can be texts, computer program code, comma separated values (CSV), etc. Likewise, pictures, audios/videos, web pages are also files.

Files stored on a computer can be accessed directly and searched for desired data. But to access data of a

© NCERT not to be republished

Reprint 2026-27

file through software, for example, to display monthly attendance report on school website, one has to write computer programs to access data from files.

Continuing the example of attendance at school, we need to store data about students and attendance in two separate files. Table 8.1 shows the contents of STUDENT file which has six columns, as detailed below:

RollNumber – Roll number of the student
SName – Name of the student
SDateofBirth – Date of birth of the student
GName – Name of the guardian
GPhone – Phone number of the student guardian
GAddress – Address of the guardian of the student

Table 8.1 STUDENT file maintained by office staff

Roll Number	SName	SDateof Birth	GName	GPhone	GAddress
1	Atharv Ahuja	2003-05-15	Amit Ahuja	5711492685	G-35, Ashok Vihar, Delhi
2	Daizy Bhutia	2002-02-28	Baichung Bhutia	7110047139	Flat no. 5, Darjeeling Appt., Shimla
3	Taleem Shah	2002-02-28	Himanshu Shah	9818184855	26/77, West Patel Nagar, Ahmedabad
4	John Dsouza	2003-08-18	Danny Dsouza		S -13, Ashok Village, Daman
5	Ali Shah	2003-07-05	Himanshu Shah	9818184855	26/77, West Patel Nagar, Ahmedabad
6	Manika P.	2002-03-10	Sujata P.	7802983674	HNO-13, B- block, Preet Vihar, Madurai

Table 8.2 shows another file called ATTENDANCE which has four columns, as detailed below:

AttendanceDate – Date for which attendance was marked
RollNumber – Roll number of the student
SName – Name of the student
AttendanceStatus – Marked as P (present) or A (absent)

Table 8.2 ATTENDANCE file maintained by class teacher

AttendanceDate	RollNumber	SName	AttendanceStatus
2018-09-01	1	Atharv Ahuja	P
2018-09-01	2	Daizy Bhutia	P
2018-09-01	3	Taleem Shah	A
2018-09-01	4	John Dsouza	P
2018-09-01	5	Ali Shah	A
2018-09-01	6	Manika P.	P

© NCERT not to be republished

Reprint 2026-27

2018-09-02	1	Atharv Ahuja	P
2018-09-02	2	Daizy Bhutia	P
2018-09-02	3	Taleem Shah	A
2018-09-02	4	John Dsouza	A
2018-09-02	5	Ali Shah	P
2018-09-02	6	Manika P.	P

8.2.1 Limitations of a File System

File system becomes difficult to handle when number of files increases and volume of data also grows. Following are some of the limitations of file system:

(A) Difficulty in Access

Files themselves do not provide any mechanism to retrieve data. Data maintained in a file system are accessed through application programs. While writing such programs, the developer may not anticipate all the possible ways in which data may be accessed. So, sometimes it is difficult to access data in the required format and one has to write application program to access data.

(B) Data Redundancy

Redundancy means same data are duplicated in different places (files). In our example, student names are maintained in both the files. Besides, in Table 8.1, students with roll numbers 3 and 5 have same guardian name and therefore same guardian name is maintained twice. Both these are examples of redundancy which is difficult to avoid in a file system. Redundancy leads to excess storage use and may cause data inconsistency also.

(C) Data Inconsistency

Data inconsistency occurs when same data maintained in different places do not match. If a student wants to get changed the spelling of her name, it needs to be changed in SName column in both the files. Likewise, if a student leaves school, the details need to be deleted from both the files. As the files are being maintained by different people, the changes may not happen in one of the files. In that case, the student name will be different (inconsistent) in both the files.

(D) Data Isolation

Both the files presented at Table 8.1 (STUDENT) and at Table 8.2 (ATTENDANCE) are related to students. But

© NCERT not to be republished

there is no link or mapping between them. The school will have to write separate programs to access these two files. This is because data mapping is not supported in file system. In a more complex system where data files are generated by different person at different times, files being created in isolation may be of different formats. In such case, it is difficult to write new application programs to retrieve data from different files maintained at multiple places, as one has to understand the underlying structure of each file as well.

(E) Data Dependence

Data are stored in a specific format or structure in a file. If the structure or format itself is changed, all the existing application programs accessing that file also need to be changed. Otherwise, the programs may not work correctly. This is data dependency. Hence, updating the structure of a data file requires modification in all the application programs accessing that file.

(F) Controlled Data Sharing

There can be different category of users like teacher, office staff and parents. Ideally, not every user should be able to access all the data. As an example, guardians and office staff can only see the student attendance data but should not be able to modify/delete it. It means these users should be given limited access (read only) to the ATTENDANCE file. Only the teacher should be able to update the attendance data. It is very difficult to enforce this kind of access control in a file system while accessing files through application programs.

8.3 Database Management System

Limitations faced in file system can be overcome by storing the data in a database where data are logically related. We can organise related data in a database so that it can be managed in an efficient and easy way.

A database management system (DBMS) or database system in short, is a software that can be used to create and manage databases. DBMS lets users to create a database, store, manage, update/modify and retrieve data from that database by users or application programs. Some examples of open source and commercial DBMS include MySQL, Oracle, PostgreSQL, SQL Server, Microsoft Access, MongoDB.

© NCERT not to be republished

Reprint 2026-27

Some database management systems include a graphical user interface for users to create and manage databases. Other database systems use a command line interface that requires users to use programming commands to create and manage databases.

A database system hides certain details about how data are actually stored and maintained. Thus, it provides users with an abstract view of the data. A database system has a set of programs through which users or other programs can access, modify and retrieve the stored data.

The DBMS serves as an interface between the database and end users or application programs. Retrieving data from a database through special type of commands is called querying the database. In addition, users can modify the structure of the database itself through a DBMS.

Databases are widely used in various fields. Some applications are given in Table 8.3.

Table 8.3 Use of Database in Real-life Applications

Application	Database to maintain data about
Banking	customer information, account details, loan details, transaction details, etc.
Crop Loan	kisan credit card data, farmer's personal data, land area and cultivation data, loan history, repayment data, etc.
Inventory Management	product details, customer information, order details, delivery data, etc.
Organisation Resource Management	employee records, salary details, department information, branch locations, etc.
Online Shopping	items description, user login details, users preferences details, etc.

8.3.1 File System to DBMS

Let us revisit our school example where two data files were maintained (Table 8.1 by office and Table 8.2 by teacher). Let us now design a database to store data of those two files. We know that tables in a database are linked or related through one or more common columns or fields. In our example, the STUDENT (Table 8.1) file and ATTENDANCE (Table 8.2) file have RollNumber and SName as common field names. In order to convert these two files into a database, we need to incorporate the following changes:

a) SName need not be maintained in ATTENDANCE file as it is already there in STUDENT. Details for a student can be retrieved through the common field RollNumber in both the files.

© NCERT not to be republished

b) If two siblings are in the same class, then same guardian details (GName, GPhone and GAddress) are maintained for both the siblings. We know this is a redundancy and by using a database we can avoid this. So let us split the STUDENT file into two file (STUDENT file and GUARDIAN) file so that each guardian data are maintained only once.

c) One and more guardians can have the same name. So it will not be possible to identify which guardian is related to which student. In such case, we need to create an additional column, say GUID (Guardian ID) that will take unique value for each record in the GUARDIAN file. The column GUID will also be kept with STUDENT file for relating these two files.

Note: We could distinguish guardians by their phone numbers also. But, phone number can change, and therefore may not truly distinguish guardian.

High Cost is incurred while shifting from file system to DBMS:

- Purchasing sophisticated hardware and software.
- Training users for querying.
- Recurrent cost to take regular backup and perform recovery operations.

Figure 8.1 shows the related data files for the STUDENT, GUARDIAN and ATTENDANCE details. Note that this is not the complete database schema since it does not show any relationship among tables.

STUDENT	GUARDIAN	ATTENDANCE
RollNumber SName SDateofBirth GUID	GUID GName GPhone GAddress	AttendanceDate RollNumber AttendanceStatus

Figure 8.1: Record structure of three files in STUDENTATTENDANCE database

The tables shown at Figure 8.1 are empty, which are to be populated with actual data as shown in Table 8.4, 8.5 and 8.6.

Table 8.4 Snapshot of STUDENT table

RollNumber	SName	SDateofBirth	GUID
1	Atharv Ahuja	2003-05-15	444444444444
2	Daizy Bhutia	2002-02-28	111111111111
3	Taleem Shah	2002-02-28	
4	John Dsouza	2003-08-18	333333333333
5	Ali Shah	2003-07-05	101010101010
6	Manika P.	2002-03-10	466444444666

© NCERT not to be republished

Reprint 2026-27

Table 8.5 Snapshot of GUARDIAN table

GUID	GName	GPhone	GAddress
444444444444	Amit Ahuja	5711492685	G-35, Ashok Vihar, Delhi
111111111111	Baichung Bhutia	3612967082	Flat no. 5, Darjeeling Appt., Shimla
101010101010	Himanshu Shah	4726309212	26/77, West Patel Nagar, Ahmedabad
333333333333	Danny Dsouza		S -13, Ashok Village, Daman
466444444666	Sujata P.	3801923168	HNO-13, B- block, Preet Vihar, Madurai

Table 8.6 Snapshot of ATTENDANCE table

Date	RollNumber	Status
2018-09-01	1	P
2018-09-01	2	P
2018-09-01	3	A
2018-09-01	4	P
2018-09-01	5	A
2018-09-01	6	P
2018-09-02	1	P
2018-09-02	2	P
2018-09-02	3	A
2018-09-02	4	A
2018-09-02	5	P
2018-09-02	6	P

Figure 8.2 shows a simplified database called STUDENTATTENDANCE, which is used to maintain data about the student, guardian and attendance. As shown here, the DBMS maintains a single repository of data at a centralised location and can be used by multiple users (office staff, teacher) at the same time.

8.3.2 Key Concepts in DBMS

In order to efficiently manage data using a DBMS, let us understand certain key terms:

(A) Database Schema

Database Schema is the design of a database. It is the skeleton of the database that represents the structure (table names and their fields/columns), the type of data each column can hold, constraints on the data to be stored (if any), and the relationships among the tables. Database schema is also called the visual or logical architecture as it tells us how the data are organised in a database.

© NCERT not to be republished

Reprint 2026-27

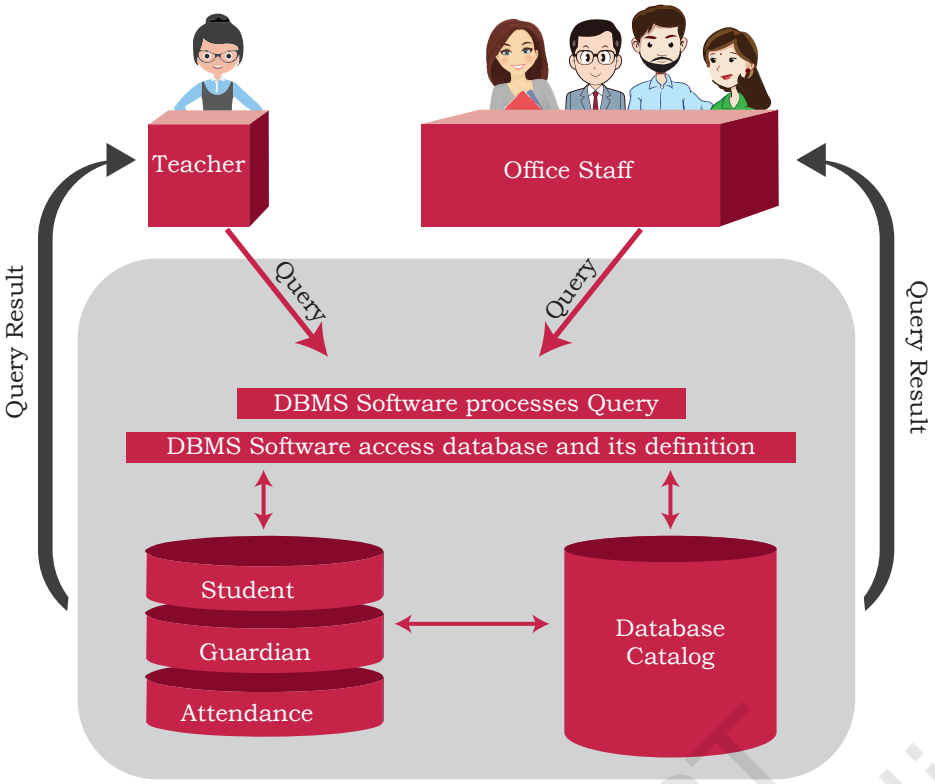


Figure 8.2: STUDENTATTENDANCE database environment

(B) Data Constraint

Sometimes we put certain restrictions or limitations on the type of data that can be inserted in one or more columns of a table. This is done by specifying one or more constraints on that column(s) while creating the tables. For example, one can define the constraint that the column mobile number can only have non-negative integer values of exactly 10 digits. Since each student shall have one unique roll number, we can put the NOT NULL and UNIQUE constraints on the RollNumber column. Constraints are used to ensure accuracy and reliability of data in the database.

(C) Meta-data or Data Dictionary

The database schema along with various constraints on the data is stored by DBMS in a database catalog or dictionary, called meta-data. A meta-data is data about the data.

(D) Database Instance

When we define database structure or schema, state of database is empty i.e. no data entry is there. After loading data, the state or snapshot of the database at any given time is the database instance. We may then retrieve data through queries or manipulate data

© NCERT not to be republished

Reprint 2026-27

through updation, modification or deletion. Thus, the state of database can change, and thus a database schema can have many instances at different times.

(E) Query

A query is a request to a database for obtaining information in a desired way. Query can be made to get data from one table or from a combination of tables. For example, "find names of all those students present on Attendance Date 2000-01-02" is a query to the database. To retrieve or manipulate data, the user needs to write query using a query language called, which is discussed in chapter 8.

(F) Data Manipulation

Modification of database consists of three operations viz. Insertion, Deletion or Update. Suppose, Rivaan joins as a new student in the class then the student details need to be added in STUDENT as well as in GUARDIAN files of the Student Attendance database. This is called Insertion operation on the database. In case a student leaves the school, then his/her data as well as her guardian details need to be removed from STUDENT, GUARDIAN and ATTENDANCE files, respectively. This is called Deletion operation on the database. Suppose Atharv's Guardian has changed his mobile number, his GPhone should be updated in GUARDIAN file. This is called Update operation on the database.

(G) Database Engine

Database engine is the underlying component or set of programs used by a DBMS to create database and handle various queries for data retrieval and manipulation.

Limitations of DBMS

Increased Complexity: Use of DBMS increases the complexity of maintaining functionalities like security, consistency, sharing and integrity

Increased data vulnerability: As data are stored centrally, it increases the chances of loss of data due to any failure of hardware or software. It can bring all operations to a halt for all the users.

8.4 Relational Data Model

Different types of DBMS are available and their classification is done based on the underlying data model. A data model describes the structure of the database, including how data are defined and represented, relationships among data, and the constraints. The most commonly used data model is Relational Data Model. Other types of data models include object-oriented data model, entity-relationship data model, document model and hierarchical data model. This book discusses the DBMS based on relational data model.

In relational model, tables are called relations that store data for different columns. Each table can have

© NCERT not to be republished

multiple columns where each column name should be unique. For example, each row in the table represents a related set of values. Each row of Table 8.5 represents a particular guardian and has related values viz. guardian's ID with guardian name, address and phone number. Thus, a table consists of a collection of relationships.

It is important to note here that relations in a database are not independent tables, but are associated with each other. For example, relation ATTENDANCE has attribute RollNumber which links it with corresponding student record in relation STUDENT. Similarly, attribute GUID is placed with STUDENT table for extracting guardian details of a particular student. If linking attributes are not there in appropriate relations, it will not be possible to keep the database in correct state and retrieve valid information from the database.

Figure 8.3 shows the relational database Student Attendance along with the three relations (tables) STUDENT, ATTENDANCE and GUARDIAN.

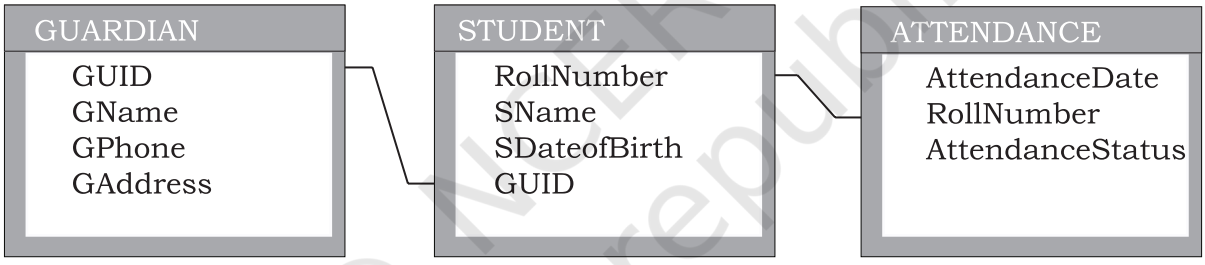


Figure 8.3: Representing STUDENTATTENDANCE database using Relational Data Model

Table 8.7 Relation schemas along with its description of Student Attendance database

Relation Scheme	Description of attributes
STUDENT(RollNumber, SName, SDateofBirth, GUID)	RollNumber: unique id of the student SName: name of the student SDateofBirth: date of birth of the student GUID: unique id of the guardian of the student
ATTENDANCE (AttendanceDate, RollNumber, AttendanceStatus)	AttendanceDate: date on which attendance is taken RollNumber: roll number of the student AttendanceStatus: whether present (P) or absent(A) Note that combination of AttendanceDate and RollNumber will be unique in each record of the table
GUARDIAN(GUID, GName, GPhone, GAddress)	GUID: unique id of the guardian GName: name of the guardian GPhone: contact number of the guardian GAddress: contact address of the guardian

© NCERT not to be republished

Reprint 2026-27

Each tuple (row) in a relation (table) corresponds to data of a real world entity (for example, Student, Guardian, and Attendance). In the GUARDIAN relation (Table 8.5), each row represents the facts about the guardian and each column name in the GUARDIAN table is used to interpret the meaning of data stored under that column. A database that is modeled on relational data model concept is called Relational Database. Figure 8.4 shows relation GUARDIAN with some populated data.

Let us now understand the commonly used terminologies in relational data model using Figure 8.4.

Relation GUARDIAN with 4 attribute/ columns

GUID	GName	GPhone	GAddress
444444444444	Amit Ahuja	5711492685	G-35, Ashok Vihar, Delhi
111111111111	Baichung Bhutia	3612967082	Flat no. 5, Darjeeling Appt., Shimla
101010101010	Himanshu Shah	4726309212	26/77, West Patel Nagar, Ahmedabad
333333333333	Danny Dsouza		S -13, Ashok Village, Daman
466444444666	Sujata P.	3801923168	HNO-13, B- block, Preet Vihar, Madurai

Relation State

Record/tuple/row

Facts about RELATION GUARDIAN:
1. Degree (Number of attributes) = 4
2. Cardinality (Number of rows/tuples/records) = 5
3. Relation is a flat file i.e, each column has a single value and each record has same number of columns

Figure 8.4: Relation GUARDIAN with its attributes and tuples

i) **ATTRIBUTE:** Characteristic or parameters for which data are to be stored in a relation. Simply stated, the columns of a relation are the attributes which are also referred as fields. For example, GUID, GName, GPhone and GAddress are attributes of relation GUARDIAN.

ii) **TUPLE:** Each row of data in a relation (table) is called a tuple. In a table with n columns, a tuple is a relationship between the n related values.

iii) **DOMAIN:** It is a set of values from which an attribute can take a value in each row. Usually, a data type is used to specify domain for an attribute. For example, in STUDENT relation, the attribute RollNumber takes integer values and hence its domain is a set of integer values. Similarly, the set of character strings constitutes the domain of the attribute SName.

iv) **DEGREE:** The number of attributes in a relation is called the Degree of the relation. For example, relation GUARDIAN with four attributes is a relation of degree 4.

© NCERT not to be republished

v) **CARDINALITY:** The number of tuples in a relation is called the Cardinality of the relation. For example, the cardinality of relation GUARDIAN is 5 as there are 5 tuples in the table.

8.4.1 Three Important Properties of a Relation

In relational data model, following three properties are observed with respect to a relation which makes a relation different from a data file or a simple table.

Property 1: imposes following rules on an attribute of the relation.

- Each attribute in a relation has a unique name.
- Sequence of attributes in a relation is immaterial.

Property 2: governs following rules on a tuple of a relation.

- Each tuple in a relation is distinct. For example, data values in no two tuples of relation ATTENDANCE can be identical for all the attributes. Thus, each tuple of a relation must be uniquely identified by its contents.
- Sequence of tuples in a relation is immaterial. The tuples are not considered to be ordered, even though they appear to be in tabular form.

Property 3: imposes following rules on the state of a relation.

- All data values in an attribute must be from the same domain (same data type).
- Each data value associated with an attribute must be atomic (cannot be further divisible into meaningful subparts). For example, GPhone of relation GUARDIAN has ten digit numbers which is indivisible.
- No attribute can have many data values in one tuple. For example, Guardian cannot specify multiple contact numbers under GPhone attribute.
- A special value "NULL" is used to represent values that are unknown or non-applicable to certain attributes. For example, if a guardian does not share his or her contact number with the school authorities, then GPhone is set to NULL (data unknown).

8.5 Keys in a Relational Database

The tuples within a relation must be distinct. It means no two tuples in a table should have same value for all attributes. That is, there should be at least one attribute

© NCERT not to be republished

Reprint 2026-27

in which data are distinct (unique) and not NULL. That way, we can uniquely distinguish each tuple of a relation. So, relational data model imposes some restrictions or constraints on the values of the attributes and how the contents of one relation be referred through another relation. These restrictions are specified at the time of defining the database through different types of keys as given below:

8.5.1 Candidate Key

A relation can have one or more attributes that takes distinct values. Any of these attributes can be used to uniquely identify the tuples in the relation. Such attributes are called candidate keys as each of them are candidates for the primary key.

As shown in Figure 8.4, the relation GUARDIAN has four attributes out of which GUID and GPhone always take unique values. No two guardians will have same phone number or same GUID. Hence, these two attributes are the candidate keys as they both are candidates for primary key.

8.5.2 Primary Key

Out of one or more candidate keys, the attribute chosen by the database designer to uniquely identify the tuples in a relation is called the primary key of that relation. The remaining attributes in the list of candidate keys are called the alternate keys.

In the relation GUARDIAN, suppose GUID is chosen as primary key, then GPhone will be called the alternate key.

8.5.3 Composite Primary Key

If no single attribute in a relation is able to uniquely distinguish the tuples, then more than one attribute are taken together as primary key. Such primary key consisting of more than one attribute is called Composite Primary key.

In relation ATTENDANCE, Roll Number cannot be used as primary key as roll number of same student will appear in another row for a different date. Similarly, in relation Attendance, AttendanceDate cannot be used as primary key because same date is repeated for each roll number. However combination of these two attributes RollNumber and AttendanceDate together would always have unique value in ATTENDANCE table as on any working day, of a student would be

© NCERT not to be republished

marked attendance only once. Hence {RollNumber, AttendanceDate} will make the of ATTENDANCE relation composite primary key.

8.5.4 Foreign Key

A foreign key is used to represent the relationship between two relations. A foreign key is an attribute whose value is derived from the primary key of another relation. This means that any attribute of a relation (referencing), which is used to refer contents from another (referenced) relation, becomes foreign key if it refers to the primary key of referenced relation. The referencing relation is called Foreign Relation. In some cases, foreign key can take NULL value if it is not the part of primary key of the foreign table. The relation in which the referenced primary key is defined is called primary relation or master relation.

In Figure 8.5, two foreign keys in Student Attendance database are shown using schema diagram where the foreign key is displayed as a directed arc (arrow) originating from it and ending at the corresponding attribute of the primary key of the referenced table. The underlined attributes make the primary key of that table.

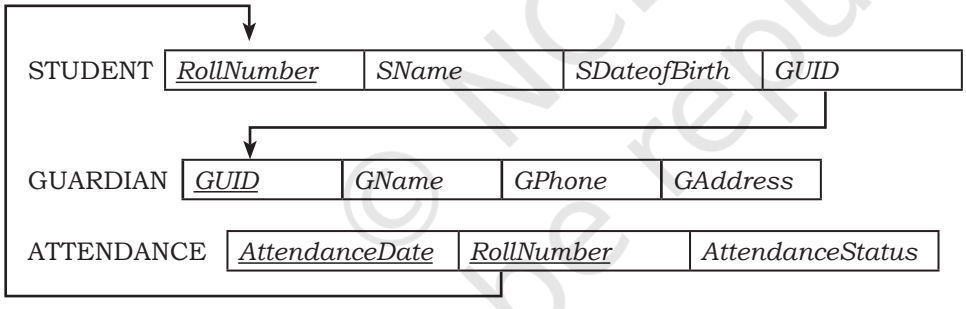


Figure 8.5: STUDENTATTENDANCE database with the primary and foreign keys

© NCERT not to be republished

Summary

- A file in a file system is a container to store data in a computer.
- File system suffers from Data Redundancy, Data Inconsistency, Data Isolation, Data Dependence and Controlled Data sharing.
- Database Management System (DBMS) is a software to create and manage databases. A database is a collection of tables.
- Database schema is the design of a database.
- A database constraint is a restriction on the type of data that that can be inserted into the table.

Reprint 2026-27

- Database schema and database constraints are stored in database catalog.
- The snapshot of the database at any given time is the database instance.
- A query is a request to a database for information retrieval and data manipulation (insertion, deletion or update). It is written in Structured Query Language (SQL).
- Relational DBMS (RDBMS) is used to store data in related tables. Rows and columns of a table are called tuples and attributed respectively. A table is referred to as a relation.
- Destructions on data stored in a RDBMS is applied by use of keys such as Candidate Key, Primary Key, Composite Primary Key, Foreign Key.
- Primary key in a relation is used for unique identification of tuples.
- Foreign key is used to relate two tables or relations.
- Each column in a table represents a feature (attribute) of a record. Table stores the information for an entity whereas a row represents a record.
- Each row in a table represents a record. A tuple is a collection of attribute values that makes a record unique.
- A tuple is a unique entity whereas attribute values can be duplicate in the table.
- SQL is the standard language for RDBMS systems like MySQL.

Exercise

1. Give the terms for each of the following:
   a) Collection of logically related records.
   b) DBMS creates a file that contains description about the data stored in the database.
   c) Attribute that can uniquely identify the tuples in a relation.
   d) Special value that is stored when actual data value is unknown for an attribute.
   e) An attribute which can uniquely identify tuples of the table but is not defined as primary key of the table.
   f) Software that is used to create, manipulate and maintain a relational database.
2. Why foreign keys are allowed to have NULL values? Explain with an example.
3. Differentiate between:
   a) Database state and database schema
   b) Primary key and foreign key
   c) Degree and cardinality of a relation

© NCERT not to be republished

Reprint 2026-27

4. Compared to a file system, how does a database management system avoid redundancy in data through a database?
5. What are the limitations of file system that can be overcome by a relational DBMS?
6. A school has a rule that each student must participate in a sports activity. So each one should give only one preference for sports activity. Suppose there are five students in a class, each having a unique roll number. The class representative has prepared a list of sports preferences as shown below. Answer the following:

Table: Sports Preferences

Roll_no	Preference
9	Cricket
13	Football
17	Badminton
17	Football
21	Hockey
24	NULL
NULL	Kabaddi

   a) Roll no 24 may not be interested in sports. Can a NULL value be assigned to that student's preference field?

   b) Roll no 17 has given two preferences in sports. Which property of relational DBMC is violated here? Can we use any constraint or key in the relational DBMS to check against such violation, if any?

   c) Kabaddi was not chosen by any student. Is it possible to have this tuple in the Sports Preferences relation?

7. In another class having 2 sections, the two respective class representatives have prepared 2 separate Sports Preferences tables, as shown below:

   Sports preference of section 1 (arranged on roll number column)

Table: Sports Preferences

Roll_no	Sports
9	Cricket
13	Football
17	Badminton
21	Hockey
24	Cricket

   Sports preference of section 2 (arranged on Sports name column, and column order is also different)

© NCERT not to be republished

Table: Sports Preferences

Sports	Roll_no
Badminton	17
Cricket	9
Cricket	24
Football	13
Hockey	21

Are the states of both the relations equivalent? Justify.

8. The school canteen wants to maintain records of items available in the school canteen and generate bills when students purchase any item from the canteen. The school wants to create a canteen database to keep track of items in the canteen and the items purchased by students. Design a database by answering the following questions:
   a) To store each item name along with its price, what relation should be used? Decide appropriate attribute names along with their data type. Each item and its price should be stored only once. What restriction should be used while defining the relation?
   b) In order to generate bill, we should know the quantity of an item purchased. Should this information be in a new relation or a part of the previous relation? If a new relation is required, decide appropriate name and data type for attributes. Also, identify appropriate primary key and foreign key so that the following two restrictions are satisfied:
      i) The same bill cannot be generated for different orders.
      ii) Bill can be generated only for available items in the canteen.
   c) The school wants to find out how many calories students intake when they order an item. In which relation should the attribute 'calories' be stored?

9. An organisation wants to create a database EMP-DEPENDENT to maintain following details about its employees and their dependent.

```
EMPLOYEE(AadharNumber, Name, Address,
        Department,EmployeeID)

DEPENDENT(EmployeeID, DependentName,
        Relationship)
```

   a) Name the attributes of EMPLOYEE, which can be used as candidate keys.
   b) The company wants to retrieve details of dependent of a particular employee. Name the tables and the key which are required to retrieve this detail.

© NCERT not to be republished

c) What is the degree of EMPLOYEE and DEPENDENT relation?

10. School uniform is available at M/s Sheetal Private Limited. They have maintained SCHOOL_UNIFORM Database with two relations viz. UNIFORM and COST. The following figure shows database schema and its state.

School Uniform Database

Attributes and Constraints

Table: UNIFORM			
Attribute	**UCode**	**UName**	**UColor**
Constraints	Primary Key	Not Null	-

Table: COST			
Attribute	**UCode**	**Size**	**Price**
Constraints	Composite Primary Key		>0

Table: UNIFORM

UCode	UName	UColor
1	Shirt	White
2	Pant	Grey
3	Skirt	Grey
4	Tie	Blue
5	Socks	Blue
6	Belt	Blue

Table: COST

UCode	Size	COST Price
1	M	500
1	L	580
1	XL	620
2	M	810
2	L	890
2	XL	940
3	M	770
3	L	830
3	XL	910
4	S	150
4	L	170
5	S	180
5	L	210
6	M	110
6	L	140
6	XL	160

a) Can they insert the following tuples to the UNIFORM Relation? Give reasons in support of your answer.

i) 7, Handkerchief, NULL
ii) 4, Ribbon, Red
iii) 8, NULL, White

b) Can they insert the following tuples to the COST Relation? Give reasons in support of your answer.

i) 7, S, 0
ii) 9, XL, 100

11. In a multiplex, movies are screened in different auditoriums. One movie can be shown in more than one auditorium. In order to maintain the record of movies, the multiplex maintains a relational database consisting of two relations viz. MOVIE and AUDI respectively as shown below:

```
Movie(Movie_ID, MovieName, ReleaseDate)
Audi(AudiNo, Movie_ID, Seats, ScreenType,
                TicketPrice)
```

© NCERT not to be republished

a) Is it correct to assign Movie_ID as the primary key in the MOVIE relation? If no, then suggest an appropriate primary key.
b) Is it correct to assign AudiNo as the primary key in the AUDI relation? If no, then suggest appropriate primary key.
c) Is there any foreign key in any of these relations?

Student Project Database

Table: STUDENT

Roll No	Name	Class	Section	Registration_ID
11	Mohan	XI	1	IP-101-15
12	Sohan	XI	2	IP-104-15
21	John	XII	1	CS-103-14
22	Meena	XII	2	CS-101-14
23	Juhi	XII	2	CS-101-10

Table: PROJECT

ProjectNo	PName	SubmissionDate
101	Airline Database	12/01/2018
102	Library Database	12/01/2018
103	Employee Database	15/01/2018
104	Student Database	12/01/2018
105	Inventory Database	15/01/2018
106	Railway Database	15/01/2018

Table: PROJECT ASSIGNED

Registration_ID	ProjectNo
IP-101-15	101
IP-104-15	103
CS-103-14	102
CS-101-14	105
CS-101-10	104

12. For the above given database STUDENT-PROJECT, answer the following:
    a) Name primary key of each table.
    b) Find foreign key(s) in table PROJECT-ASSIGNED.
    c) Is there any alternate key in table STUDENT? Give justification for your answer.
    d) Can a user assign duplicate value to the field RollNo of STUDENT table? Jusify.

13. For the above given database STUDENT-PROJECT, can we perform the following operations?
    a) Insert a student record with missing roll number value.
    b) Insert a student record with missing registration number value.
    c) Insert a project detail without submission-date.
    d) Insert a record with registration ID IP-101-19 and ProjectNo 206 in table PROJECT-ASSIGNED.

© NCERT not to be republished

Reprint 2026-27

Chapter 9

Structured Query Language (SQL)

12130CH09

"Any unique image that you desire probably already exists on the internet or in some database... The problem today is no longer how to create the right image, but how to find an already existing one."

— Lev Manovich

In this Chapter

- *Introduction*
- *Structured Query Language (SQL)*
- *Data Types and Constraints in MySQL*
- *SQL for Data Definition*
- *SQL for Data Manipulation*
- *SQL for Data Query*
- *Data Updation and Deletion*
- *Functions in SQL*
- *GROUP BY Clause in SQL*
- *Operations on Relations*
- *Using Two Relations in a Query*

9.1 Introduction

We have learnt about Relational Database Management Systems (RDBMS) and its purpose in the previous chapter. There are many RDBMS such as MySQL, Microsoft SQL Server, PostgreSQL, Oracle, etc. that allow us to create a database consisting of relations. These RDBMS also allow us to store, retrieve and manipulate data on that database through queries. In this chapter, we will learn how to create, populate and query databases using MySQL.

9.2 Structured Query Language (SQL)

One has to write application programs to access data in case of a file system. However, for database management systems there are special kinds of languages called query language that can be used to access and manipulate data from the database. The Structured Query Language (SQL) is the most popular query language used by major relational

© NCERT not to be republished

database management systems such as MySQL, ORACLE, SQL Server, etc.

Activity 9.1

Find and list other types of databases other than RDBMS.

SQL is easy to learn as the statements comprise of descriptive English words and are not case sensitive. We can create and interact with a database using SQL easily. Benefit of using SQL is that we do not have to specify how to get the data from the database. Rather, we simply specify what is to be retrieved, and SQL does the rest. Although called a query language, SQL can do much more, besides querying. SQL provides statements for defining the structure of the data, manipulating data in the database, declaring constraints and retrieving data from the database in various ways, depending on our requirements.

In this chapter, we will use the StudentAttendance discussed in chapter 8 and create a database. We will also learn how to populate databases with data, manipulate data and retrieve data from a database through SQL queries.

9.2.1 Installing MySQL

MySQL is an open source RDBMS software which can be easily downloaded from the official website https://dev.mysql.com/downloads. After installing MySQL, start MySQL service. The appearance of mysql> prompt (Figure 9.1) means that MySQL is ready to accept SQL statements.

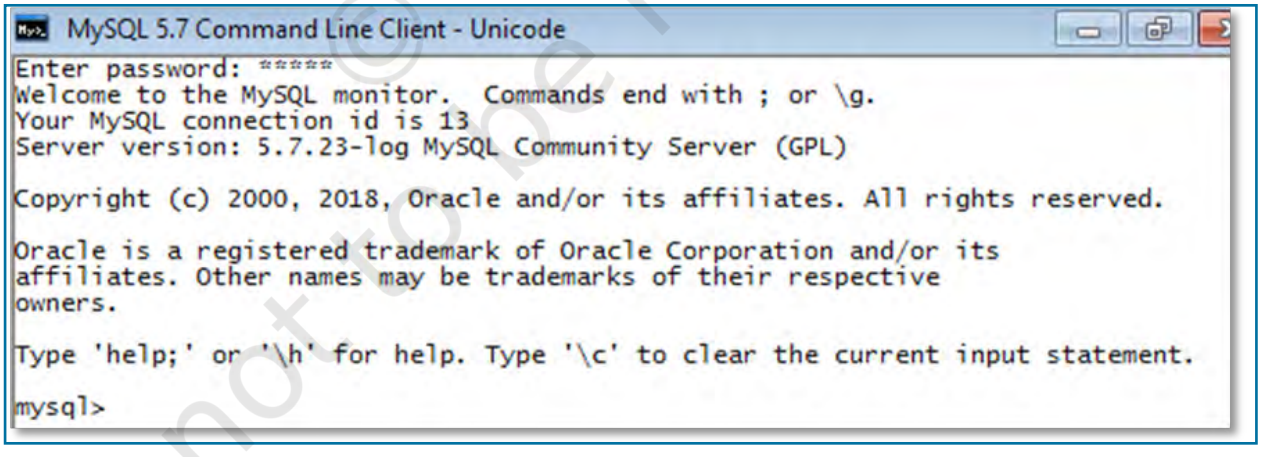


Figure 9.1: MySQL Shell

Following are some important points to be kept in mind while using SQL:

© NCERT not to be republished

- SQL is case insensitive. For example, the column names 'salary' and 'SALARY' are the same for SQL.
- Always end SQL statements with a semicolon (;).
- To enter multiline SQL statements, we don't write ";" after the first line. We press the Enter key to continue on the next line. The prompt mysql> then changes to "->", indicating that statement is continued to the next line. After the last line, put ";" and press enter.

9.3 Data Types and Constraints in MySQL

We know that a database consists of one or more relations and each relation (table) is made up of attributes (column). Each attribute has a data type. We can also specify constraints for each attribute of a relation.

9.3.1 Data type of Attribute

Data type of an attribute indicates the type of data value that an attribute can have. It also decides the operations that can be performed on the data of that attribute. For example, arithmetic operations can be performed on numeric data but not on character data. Commonly used data types in MySQL are numeric types, date and time types, and string types as shown in Table 9.1.

Activity 9.2

What are the other data types supported in MySQL? Are there other variants of integer and float data type?

Table 9.1 Commonly used data types in MySQL

Data type	Description
CHAR(n)	Specifies character type data of length n where n could be any value from 0 to 255. CHAR is of fixed length, means, declaring CHAR (10) implies to reserve spaces for 10 characters. If data does not have 10 characters (e.g., 'city' has four characters), MySQL fills the remaining 6 characters with spaces padded on the right.
VARCHAR(n)	Specifies character type data of length where n could be any value from 0 to 65535. But unlike CHAR, VARCHAR(n) is a variable-length data type. That is, declaring VARCHAR (30) means a maximum of 30 characters can be stored but the actual allocated bytes will depend on the length of entered string. So 'city' in VARCHAR (30) will occupy space needed to store 4 characters only.
INT	INT specifies an integer value. Each INT value occupies 4 bytes of storage. The range of unsigned values allowed in a 4 byte integer type are 0 to 4,294,967,295. For values larger than that, we have to use BIGINT, which occupies 8 bytes.
FLOAT	Holds numbers with decimal points. Each FLOAT value occupies 4 bytes.
DATE	The DATE type is used for dates in 'YYYY-MM-DD' format. YYYY is the 4 digit year, MM is the 2 digit month and DD is the 2 digit date. The supported range is '1000-01-01' to '9999-12-31'.

© NCERT not to be republished

Reprint 2026-27

Think and Reflect

Which two constraints when applied together will produce a Primary Key constraint?

9.3.2 Constraints

Constraints are the certain types of restrictions on the data values that an attribute can have. Table 9.2 lists some of the commonly used constraints in SQL. They are used to ensure correctness of data. However, it is not mandatory to define constraints for each attribute of a table.

Table 9.2 Commonly used SQL Constraints

Constraint	Description
NOT NULL	Ensures that a column cannot have NULL values where NULL means missing/ unknown/not applicable value.
UNIQUE	Ensures that all the values in a column are distinct/unique
DEFAULT	A default value specified for the column if no value is provided
PRIMARY KEY	The column which can uniquely identify each row/record in a table.
FOREIGN KEY	The column which refers to value of an attribute defined as primary key in another table

9.4 SQL for Data Definition

In order to be able to store data we need to first define the relation schema. Defining a schema includes creating a relation and giving name to a relation, identifying the attributes in a relation, deciding upon the datatype for each attribute and also specify the constraints as per the requirements. Sometimes, we may require to make changes to the relation schema also. SQL allows us to write statements for defining, modifying and deleting relation schemas. These are part of Data Definition Language (DDL).

We have already learned that the data are stored in relations or tables in a database. Hence, we can say that a database is a collection of tables. The Create statement is used to create a database and its tables (relations). Before creating a database, we should be clear about the number of tables the database will have, the columns (attributes) in each table along with the data type of each column, and its constraint, if any.

9.4.1 CREATE Database

To create a database, we use the CREATE DATABASE statement as shown in the following syntax:

```
CREATE DATABASE databasename;
```

To create a database called `StudentAttendance`, we will type following command at mysql prompt.

© NCERT not to be republished

```
mysql> CREATE DATABASE StudentAttendance;
Query OK, 1 row affected (0.02 sec)
```

Note: In LINUX environment, names for database and tables are case-sensitive whereas in WINDOWS, there is no such differentiation. However, as a good practice, it is suggested to write database/table name in the same letter cases that were used at the time of their creation.

A DBMS can manage multiple databases on one computer. Therefore, we need to select the database that we want to use. To know the names of existing databases, we use the statement SHOW DATABASES. From the listed databases, we can select the database to be used. Once the database is selected, we can proceed with creating tables or querying data.
In order to use the StudentAttendance database, the following SQL statement is required.

```
mysql> USE StudentAttendance;
Database changed
```

Initially, the created database is empty. It can be checked by using the show tables statement that lists names of all the tables within a database.

```
mysql> SHOW TABLES;
Empty set (0.06 sec)
```

Activity 9.3

Type the statement show database; Does it show the name of StudentAttendance database?

9.4.2 CREATE Table

After creating a database StudentAttendance, we need to define relations in this database and specify attributes for each relation along with data type and constraint (if any) for each attribute. This is done using the CREATE TABLE statement.

Syntax:

```
CREATE TABLE tablename(
attributename1 datatype constraint,
attributename2 datatype constraint,
:
attributenameN datatype constraint);
```

It is important to observe the following points with respect to the CREATE TABLE statement:

- The number of columns in a table defines the degree of that relation, which is denoted by N.
- Attribute name specifies the name of the column in the table.
- Datatype specifies the type of data that an attribute can hold.

© NCERT not to be republished

Reprint 2026-27

- Constraint indicates the restrictions imposed on the values of an attribute. By default, each attribute can take NULL values except for the primary key.

Let us identify data types of the attributes of table STUDENT along with their constraints (if any). Assuming maximum students in a class to be 100 and values of roll number in a sequence from 1 to 100, we know that 3 digits are sufficient to store values for the attribute RollNumber. Hence, data type INT is appropriate for this attribute. Total number of characters in a student name (SName) can differ. Assuming maximum characters in a name as 20, we use VARCHAR(20) for the SName column. Data type for the attribute SDateofBirth is DATE and supposing the school uses guardian's 12 digit Aadhaar number as GUID, we can declare GUID as CHAR (12) since Aadhaar number is of fixed length and we are not going to perform any mathematical operation on GUID.

Table 9.3, 9.4 and 9.5 shows the chosen data type and constraint for each attribute of the relations STUDENT, GUARDIAN and ATTENDANCE, respectively.

Table 9.3 Data types and constraints for the attributes of relation STUDENT

Attribute Name	Data expected to be stored	Data type	Constraint
RollNumber	Numeric value consisting of maximum 3 digits	INT	PRIMARY KEY
SName	Variant length string of maximum 20 characters	VARCHAR(20)	NOT NULL
SDateofBirth	Date value	DATE	NOT NULL
GUID	Numeric value consisting of 12 digits	CHAR (12)	FOREIGN KEY

Table 9.4 Data types and constraints for the attributes of relation GUARDIAN

Attribute Name	Data expected to be stored	Data type	Constraint
GUID	Numeric value consisting of 12 digit Aadhaar number	CHAR (12)	PRIMARY KEY
GName	Variant length string of maximum 20 characters	VARCHAR(20)	NOT NULL
GPhone	Numeric value consisting of 10 digits	CHAR(10)	NULL UNIQUE
GAddress	Variant length String of size 30 characters	VARCHAR(30)	NOT NULL

Table 9.5 Data types and constraints for the attributes of relation ATTENDANCE.

Attribute Name	Data expected to be stored	Data type	Constraint
AttendanceDate	Date value	DATE	PRIMARY KEY*
RollNumber	Numeric value consisting of maximum 3 digits	INT	PRIMARY KEY* FOREIGN KEY
AttendanceStatus	'P' for present and 'A' for absent	CHAR(1)	NOT NULL

means part of composite primary key.

© NCERT not to be republished

Once data types and constraints are identified, let us create tables without specifying constraints along with the attribute name for simplification. We will learn to incorporate constraints on attributes in Section 9.4.4.

Example 9.1 Create table STUDENT.

```
mysql> CREATE TABLE STUDENT(
    -> RollNumber  INT,
    -> SName VARCHAR(20),
    -> SDateofBirth DATE,
    -> GUID CHAR (12),
    -> PRIMARY KEY (RollNumber));
Query OK, 0 rows affected (0.91 sec)
```

Note: "," is used to separate two attributes and each statement terminates with a semi-colon (;). The arrow (->) is an interactive continuation prompt. If we enter an unfinished statement, the SQL shell will wait for us to enter the rest of the statement.

9.4.3 Describe Table

We can view the structure of an already created table using the DESCRIBE statement or DESC statement.

Syntax:

```
DESCRIBE tablename;
mysql> DESCRIBE STUDENT;
    +--------------+-------------+------+-----+---------+-------+
    | Field        | Type        | Null | Key | Default | Extra |
    +--------------+-------------+------+-----+---------+-------+
    | RollNumber   | int         | NO   | PRI | NULL    |       |
    | SName        | varchar(20) | YES  |     | NULL    |       |
    | SDateofBirth | date        | YES  |     | NULL    |       |
    | GUID         | char(12)    | YES  |     | NULL    |       |
    +--------------+-------------+------+-----+---------+-------+

    4 rows in set (0.06 sec)
```

We can use the SHOW TABLES statement to see the tables in the StudentAttendance database. So far, we have only the STUDENT table.

```
mysql> SHOW TABLES;
    +-----------------------------+
    | Tables_in_studentattendance |
    +-----------------------------+
    | student                     |


    +-----------------------------+
    1 row in set (0.00 sec)
```

9.4.4 ALTER Table

After creating a table, we may realise that we need to add/remove an attribute or to modify the datatype of an existing attribute or to add constraint in attribute. In

Think and Reflect

Which datatype out of Char and Varchar will you prefer for storing contact number(mobile number)? Discuss.

Activity 9.4

Create the other two relations GUARDIAN and ATTENDANCE as per data types given in Table 9.4 and 9.5 respectively, and view their structures. Do not add any constraint in these two tables.

© NCERT not to be republished

all such cases, we need to change or alter the structure (schema) of the table by using the alter statement.

(A) Add primary key to a relation

Let us now alter the tables created in Activity 9.4. The following MySQL statement adds a primary key to the GUARDIAN relation:

```
mysql> ALTER TABLE GUARDIAN ADD PRIMARY KEY (GUID);
    Query OK, 0 rows affected (1.14 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

Now let us add the primary key to the ATTENDANCE relation. The primary key of this relation is a composite key made up of two attributes - AttendanceDate and RollNumber.

```
mysql> ALTER TABLE ATTENDANCE
        -> ADD PRIMARY KEY(AttendanceDate,
    RollNumber);
    Query OK, 0 rows affected (0.52 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

Activity 9.5

Add foreign key in the ATTENDANCE table (use Figure 9.1) to identify referencing and referenced tables).

(B) Add foreign key to a relation

Once primary keys are added, the next step is to add foreign keys to the relation (if any). Following points need to be observed while adding foreign key to a relation:

- The referenced relation must be already created.
- The referenced attribute(s) must be part of the primary key of the referenced relation.
- Data types and size of referenced and referencing attributes must be the same.

Syntax:

```
ALTER TABLE table_name ADD FOREIGN KEY(attribute
name) REFERENCES referenced_table_name
(attribute name);
```

Think and Reflect

Name foreign keys in table ATTENDANCE and STUDENT. Is there any foreign key in table GUARDIAN.

Let us now add foreign key to the table STUDENT. Table 9.3 shows that attribute GUID (the referencing attribute) is a foreign key and it refers to attribute GUID (the referenced attribute) of table GUARDIAN. Hence, STUDENT is the referencing table and GUARDIAN is the referenced table as shown in Figure 8.4 in the previous chapter.

```
mysql> ALTER TABLE STUDENT
        -> ADD FOREIGN KEY(GUID) REFERENCES
        -> GUARDIAN(GUID);
    Query OK, 0 rows affected (0.75 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

(C) Add constraint UNIQUE to an existing attribute

In GUARDIAN table, the attribute GPhone has a constraint UNIQUE which means no two values in that column should be the same.

© NCERT not to be republished

Syntax:

```
ALTER TABLE table_name ADD UNIQUE (attribute
name);
```

Let us now add the constraint UNIQUE with the attribute GPhone of the table GUARDIAN as shown at table 9.4.

```
mysql> ALTER TABLE GUARDIAN
        -> ADD UNIQUE(GPhone);
    Query OK, 0 rows affected (0.44 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

(D) Add an attribute to an existing table

Sometimes, we may need to add an additional attribute in a table. It can be done using the ADD attribute statement as shown in the following Syntax:

```
ALTER TABLE table_name ADD attribute
name DATATYPE;
```

Suppose, the principal of the school has decided to award scholarship to some needy students for which income of the guardian must be known. But, the school has not maintained the income attribute with table GUARDIAN so far. Therefore, the database designer now needs to add a new attribute Income of data type INT in the table GUARDIAN.

```
mysql> ALTER TABLE GUARDIAN
        -> ADD income INT;
    Query OK, 0 rows affected (0.47 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

(E) Modify datatype of an attribute

We can change data types of the existing attributes of a table using the following ALTER statement.

Syntax:

```
ALTER TABLE table_name MODIFY attribute DATATYPE;
```

Suppose we need to change the size of the attribute GAddress from VARCHAR(30) to VARCHAR(40) of the GUARDIAN table. The MySQL statement will be:

```
mysql> ALTER TABLE GUARDIAN
    -> MODIFY GAddress VARCHAR(40);
    Query OK, 0 rows affected (0.11 sec)
    Records: 0 Duplicates: 0 Warnings: 0
```

(F) Modify constraint of an attribute

When we create a table, by default each attribute takes NULL value except for the attribute defined as primary key. We can change an attribute's constraint from NULL to NOT NULL using an alter statement.

© NCERT not to be republished

Think and Reflect

What are the minimum and maximum income values that can be entered in the income attribute given the data type is INT?

Syntax:

```
ALTER TABLE table_name MODIFY attribute DATATYPE
NOT NULL;
```

Note: We have to specify the data type of the attribute along with constraint NOT NULL while using MODIFY.

To associate NOT NULL constraint with attribute SName of table STUDENT (table 9.3), we write the following MySQL statement:

```
mysql> ALTER TABLE STUDENT
    -> MODIFY SName VARCHAR(20) NOT NULL;
Query OK, 0 rows affected (0.47 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

(G) Add default value to an attribute

If we want to specify default value for an attribute, then use the following syntax:

```
ALTER TABLE table_name MODIFY attribute DATATYPE
DEFAULT default_value;
```

To set default value of SDateofBirth of STUDENT to 15th May 2000, write the following statement:

```
mysql> ALTER TABLE STUDENT
    -> MODIFY SDateofBirth DATE DEFAULT '2000-05-
       15';
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Note: We have to specify the data type of the attribute along with DEFAULT while using MODIFY.

(H) Remove an attribute

Using ALTER, we can remove attributes from a table, as shown in the following syntax:

```
ALTER TABLE table_name DROP attribute;
To remove the attribute income from table
GUARDIAN (Table 9.4), write the following MySQL
statement:
mysql> ALTER TABLE GUARDIAN DROP income;
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

(I) Remove primary key from the table

Sometime there may be a requirement to remove primary key constraint from the table. In that case, Alter table command can be used in the following way:

Syntax:

```
ALTER TABLE table_name DROP PRIMARY KEY;
```

To remove primary key of table GUARDIAN (Figure 9.4), write the following MySQL statement:

```
mysql> ALTER TABLE GUARDIAN DROP PRIMARY KEY;
Query OK, 0 rows affected (0.72 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

© NCERT not to be republished

Reprint 2026-27

Note: We have dropped the primary key from the GUARDIAN table, but each table should have a primary key to maintain uniqueness. Hence, we have to use the ADD statement with the Alter Table command to specify the primary key for the GUARDIAN table as shown in earlier examples.

9.4.5 DROP Statement

Sometimes a table in a database or the database itself needs to be removed. We can use a DROP statement to remove a database or a table permanently from the system. However, one should be very cautious while using this statement as it cannot be undone.

Syntax to drop a table:

```
DROP TABLE table_name;
```

Syntax to drop a database:

```
DROP DATABASE database_name;
```

Note: Using the DROP statement to remove a database will ultimately remove all the tables within it.

9.5 SQL for Data Manipulation

In the previous section, we created the database StudentAttendance having three relations STUDENT, GUARDIAN and ATTENDANCE. When we create a table, only its structure is created but the table has no data. To populate records in the table, INSERT statement is used. Also, table records can be deleted or updated using DELETE and UPDATE statements. These SQL statements are part of Data Manipulation Language (DML).

Data Manipulation using a database means either insertion of new data, removal of existing data or modification of existing data in the database

9.5.1 INSERTION of Records

`INSERT INTO` statement is used to insert new records in a table. Its syntax is:

```
INSERT INTO tablename
VALUES(value 1, value 2,....);
```

Here, value 1 corresponds to attribute 1, value 2 corresponds to attribute 2 and so on. Note that we need not to specify attribute names in the insert statement if there are exactly the same numbers of values in the INSERT statement as the total number of attributes in the table.

Caution: While populating records in a table with foreign key, ensure that records in referenced tables are already populated.

© NCERT not to be republished

Let us insert some records in the StudentAttendance database. We shall insert records in the GUARDIAN table first as it does not have any foreign key. A set of sample records for GUARDIAN table is shown in the given table (Table 9.6).

Table 9.6 GUARDIAN Table

GUID	GName	GPhone	GAddress
444444444444	Amit Ahuja	5711492685	G-35, Ashok Vihar, Delhi
111111111111	Baichung Bhutia	3612967082	Flat no. 5, Darjeeling Appt., Shimla
101010101010	Himanshu Shah	4726309212	26/77, West Patel Nagar, Ahmedabad
333333333333	Danny Dsouza		S -13, Ashok Village, Daman
466444444666	Sujata P.	3801923168	HNO-13, B- block, Preet Vihar, Madurai

The following insert statement adds the first record in the table:

```
mysql> INSERT INTO GUARDIAN
    -> VALUES (444444444444, 'Amit Ahuja',
    -> 5711492685, 'G-35,Ashok vihar, Delhi' );
    Query OK, 1 row affected (0.01 sec)
```

We can use the SQL statement SELECT * from table_name to view the inserted records. The SELECT statement will be explained in the next section.

```
mysql> SELECT * from  GUARDIAN;
+--------------+---------------+-----------+--------------------------+
| GUID         | GName         | Gphone    | GAddress                 |
+--------------+---------------+-----------+--------------------------+
| 444444444444 | Amit Ahuja    | 5711492685 | G-35, Ashok vihar, Delhi  |
+--------------+---------------+-----------+--------------------------+
    1 row in set (0.00 sec)
```

If we want to insert values only for some of the attributes in a table (supposing other attributes having NULL or any other default value), then we shall specify the attribute names in which the values are to be inserted using the following syntax of INSERT INTO statement.

Syntax:

```
INSERT INTO tablename (column1, column2, ...)
VALUES (value1, value2,  ...);
```

To insert the fourth record of Table 9.6 where GPhone is not given, we need to insert values in the other three fields (GPhone was set to NULL by default at the time of table creation). In this case, we have to specify the names of attributes in which we want to insert values. The values must be given in the same order in which attributes are written in INSERT statement.

Activity 9.6

Write SQL statements to insert the remaining 3 rows of table 9.6 in table GUARDIAN.

© NCERT not to be republished

```
mysql> INSERT INTO GUARDIAN(GUID, GName, GAddress)
    -> VALUES (333333333333, 'Danny Dsouza',
    -> 'S -13, Ashok Village, Daman' );
    Query OK, 1 row affected (0.03 sec)
```

Note: Text and date values must be enclosed in ‘ ’ (single quotes).

```
mysql> SELECT * from GUARDIAN;
    +--------------+--------------+-----------+--------------------------+
    | GUID         | GName        | Gphone    | GAddress                 |
    +--------------+--------------+-----------+--------------------------+
    | 333333333333 | Danny Dsouza | NULL      | S -13, Ashok Village,Daman|
    | 444444444444 | Amit Ahuja   | 5711492685| G-35, Ashok vihar, Delhi  |
    +--------------+--------------+-----------+--------------------------+
    2 rows in set (0.00 sec)
```

Let us now insert the records given in Table 9.7 into the STUDENT table.

Table 9.7 STUDENT Table

RollNumber	SName	SDateofBirth	GUID
1	Atharv Ahuja	2003-05-15	444444444444
2	Daizy Bhutia	2002-02-28	111111111111
3	Taleem Shah	2002-02-28	
4	John Dsouza	2003-08-18	333333333333
5	Ali Shah	2003-07-05	101010101010
6	Manika P.	2002-03-10	466444444666

To insert the first record of Table 9.7, we write the following MySQL statement

```
mysql> INSERT INTO STUDENT
    -> VALUES(1,'Atharv Ahuja','2003-05-15',
    444444444444);
    Query OK, 1 row affected (0.11 sec)
    OR
    mysql> INSERT INTO STUDENT (RollNumber, SName,
    SDateofBirth, GUID)
        -> VALUES (1,'Atharv Ahuja','2003-05-15',
    444444444444);
    Query OK, 1 row affected (0.02 sec)
```

Activity 9.7

Write SQL statements to insert the remaining 4 rows of table 9.7 in table STUDENT.

Recall that Date is stored in ‘YYYY-MM-DD’ format.

```
mysql> SELECT * from STUDENT;
    +------------+--------------+--------------+--------------+
    | RollNumber | SName        | SDateofBirth | GUID         |
    +------------+--------------+--------------+--------------+
    |          1 | Atharv Ahuja | 2003-05-15   | 444444444444 |
    +------------+--------------+--------------+--------------+
    1 row in set (0.00 sec)
```

Let us now insert the third record of Table 9.7 where GUID is NULL. Recall that GUID is foreign key of this table and thus can take NULL value. Hence, we can put NULL value for GUID and insert the record by using the following statement:

© NCERT not to be republished

Reprint 2026-27

```
mysql> INSERT INTO STUDENT
    -> VALUES(3, 'Taleem Shah','2002-02-28', NULL);
    Query OK, 1 row affected (0.05 sec)

mysql> SELECT * from STUDENT;
+------------+--------------+--------------+--------------+
| RollNumber | SName        | SDateofBirth | GUID         |
+------------+--------------+--------------+--------------+
|          1 | Atharv Ahuja | 2003-05-15   | 444444444444 |
|          3 | Taleem Shah  | 2002-02-28   |         NULL |
+------------+--------------+--------------+--------------+
    2 rows in set (0.00 sec)
```

We had to write NULL in the above insert statement because we are not mentioning the column names. Otherwise, we should mention the names of attributes along with the values if we need to insert data only for certain attributes, as shown in the following query:

```
mysql> INSERT INTO STUDENT (RollNumber, SName,
-> SDateofBirth) VALUES (3, 'Taleem Shah','2002-02-
28');
Query OK, 1 row affected (0.05 sec)
```

Think and Reflect

- Which of the two insert statement should be used when the order of data to be inserted are not known?
- Can we insert two records with the same roll number?

9.6 SQL for Data Query

So far we have learnt how to create a database and how to store and manipulate data in them. We are interested in storing data in a database as it is easier to retrieve data in future from databases in whatever way we want. SQL provides efficient mechanisms to retrieve data stored in multiple tables in MySQL database (or any other RDBMS). The SQL statement SELECT is used to retrieve data from the tables in a database and is also called a query statement.

9.6.1 SELECT Statement

The SQL statement SELECT is used to retrieve data from the tables in a database and the output is also displayed in tabular form.

Syntax:

```
SELECT attribute1, attribute2, ...
FROM table_name
WHERE condition;
```

Here, attribute1, attribute2, ... are the column names of the table table_name from which we want to retrieve data. The FROM clause is always written with SELECT clause as it specifies the name of the table from which data is to be retrieved. The WHERE clause is optional and is used to retrieve data that meet specified condition(s).

© NCERT not to be republished

To select all the data available in a table, we use the following select statement:

```
SELECT * FROM table_name;
```

Example 9.2 The following query retrieves the name and date of birth of student with roll number 1:

```
mysql> SELECT SName, SDateofBirth
    -> FROM STUDENT
    -> WHERE RollNumber = 1;
    +--------------+--------------+
    | SName        | SDateofBirth |
    +--------------+--------------+
    | Atharv Ahuja | 2003-05-15   |
    +--------------+--------------+
    1 row in set (0.03 sec)
```

Think and Reflect

Think and list few examples from your daily life where storing the data in the database and querying the same can be helpful.

9.6.2 QUERYING using Database OFFICE

Organisations maintain databases to store data in the form of tables. Let us consider the database OFFICE of an organisation that has many related tables like EMPLOYEE, DEPARTMENT and so on. Every EMPLOYEE in the database is assigned to a DEPARTMENT and his/her Department number (DeptId) is stored as a foreign key in the table EMPLOYEE. Let us consider the relation 'EMPLOYEE' as shown in Table 9.8 and apply the SELECT statement to retrieve data:

Table 9.8 Records to be inserted into the EMPLOYEE table

EmpNo	Ename	Salary	Bonus	DeptId
101	Aaliya	10000	234	D02
102	Kritika	60000	123	D01
103	Shabbbir	45000	566	D01
104	Gurpreet	19000	565	D04
105	Joseph	34000	875	D03
106	Sanya	48000	695	D02
107	Vergese	15000		D01
108	Nachaobi	29000		D05
109	Daribha	42000		D04
110	Tanya	50000	467	D05

(A) Retrieve selected columns

The following query selects employee numbers of all the employees:

```
mysql> SELECT  EmpNo FROM  EMPLOYEE;
    +-------+
    | EmpNo |
    +-------+
    |   101 |
    |   102 |
```

© NCERT not to be republished

```
|   103 |
|   104 |
|   105 |
|   106 |
|   107 |
|   108 |
|   109 |
|   110 |
+-------+
10 rows in set (0.41 sec)
```

The following query selects the employee number and employee name of all the employees, we write:

```
mysql> SELECT EmpNo, Ename FROM  EMPLOYEE;
+-------+----------+
| EmpNo | Ename    |
+-------+----------+
|   101 | Aaliya   |
|   102 | Kritika  |
|   103 | Shabbir  |
|   104 | Gurpreet |
|   105 | Joseph   |
|   106 | Sanya    |
|   107 | Vergese  |
|   108 | Nachaobi |
|   109 | Daribha  |
|   110 | Tanya    |
+-------+----------+
10 rows in set (0.00 sec)
```

(B) Renaming of columns

In case we want to rename any column while displaying the output, it can be done by using the alias 'AS'. The following query selects Employee name as Name in the output for all the employees:

```
mysql> SELECT  EName  as Name FROM  EMPLOYEE;
+----------+
| Name     |
+----------+
| Aaliya   |
| Kritika  |
| Shabbir  |
| Gurpreet |
| Joseph   |
| Sanya    |
| Vergese  |
| Nachaobi |
| Daribha  |
| Tanya    |
+----------+
10 rows in set (0.00 sec)
```

Example 9.3 Select names of all employees along with their annual income (calculated as Salary*12). While displaying the query result, rename the column EName as Name

```
mysql> SELECT EName as Name, Salary*12 FROM EMPLOYEE;
```

© NCERT not to be republished

```
+----------+-----------+
| Name     | Salary*12 |
+----------+-----------+
| Aaliya   |    120000 |
| Kritika  |    720000 |
| Shabbir  |    540000 |
| Gurpreet |    228000 |
| Joseph   |    408000 |
| Sanya    |    576000 |
| Vergese  |    180000 |
| Nachaobi |    348000 |
| Daribha  |    504000 |
| Tanya    |    600000 |
+----------+-----------+
10 rows in set (0.02 sec)
```

Observe that in the output, Salary*12 is displayed as the column name for the Annual Income column. In the output table, we can use alias to rename that column as Annual Income as shown below:

```
mysql> SELECT Ename AS Name, Salary*12 AS 'Annual
Income'
    -> FROM EMPLOYEE;
+----------+---------------+
| Name     | Annual Income|
+----------+---------------+
| Aaliya   |        120000 |
| Kritika  |        720000 |
| Shabbir  |        540000 |
| Gurpreet |        228000 |
| Joseph   |        408000 |
| Sanya    |        576000 |
| Vergese  |        180000 |
| Nachaobi |        348000 |
| Daribha  |        504000 |
| Tanya    |        600000 |
+----------+---------------+
10 rows in set (0.00 sec)
```

Note: Annual Income will not be added as a new column in the database table. It is just for displaying the output of the query.

If an aliased column name has space as in the case of Annual Income, it should be enclosed in quotes as 'Annual Income

(C) Distinct Clause

By default, SQL shows all the data retrieved through query as output. However, there can be duplicate values. The SELECT statement when combined with DISTINCT clause, returns records without repetition (distinct records). For example, while retrieving a department number from employee relation, there can be duplicate values as many employees are assigned to the same department. To select unique department number for all the employees, we use DISTINCT as shown below:

```
mysql> SELECT DISTINCT DeptId FROM EMPLOYEE;
```

© NCERT not to be republished

```
+--------+
| DeptId |
+--------+
| D02    |
| D01    |
| D04    |
| D03    |
| D05    |
+--------+
5 rows in set (0.03 sec)
```

(D) WHERE Clause

The WHERE clause is used to retrieve data that meet some specified conditions. In the OFFICE database, more than one employee can have the same salary. Following query gives distinct salaries of the employees working in the department number D01:

```
mysql> SELECT DISTINCT Salary
    -> FROM EMPLOYEE
    -> WHERE Deptid='D01';
```

As the column DeptId is of string type, its values are enclosed in quotes ('D01').

```
+--------+
| Salary |
+--------+
|  60000 |
|  45000 |
|  15000 |
+--------+
3 rows in set (0.02 sec)
```

In the above example, = operator is used in the WHERE clause. Other relational operators (<, <=, >, >=, !=) can be used to specify such conditions. The logical operators AND, OR, and NOT are used to combine multiple conditions.

Example 9.4 Display all the details of those employees of D04 department who earn more than 5000.

```
mysql> SELECT  * FROM  EMPLOYEE
    -> WHERE Salary > 5000 AND DeptId = 'D04';
+-------+----------+--------+-------+--------+
| EmpNo | Ename    | Salary | Bonus | DeptId |
+-------+----------+--------+-------+--------+
|   104 | Gurpreet |  19000 |   565 | D04    |
|   109 | Daribha  |  42000 |  NULL | D04    |
+-------+----------+--------+-------+--------+
2 rows in set (0.00 sec)
```

Example 9.5 The following query selects records of all the employees except Aaliya.

```
mysql> SELECT * FROM EMPLOYEE
    -> WHERE NOT Ename = 'Aaliya';
+-------+----------+--------+-------+--------+
```

© NCERT not to be republished

```
| EmpNo | Ename    | Salary | Bonus | DeptId |
+-------+----------+--------+-------+--------+
|   102 | Kritika  |  60000 |   123 | D01    |
|   103 | Shabbir  |  45000 |   566 | D01    |
|   104 | Gurpreet |  19000 |   565 | D04    |
|   105 | Joseph   |  34000 |   875 | D03    |
|   106 | Sanya    |  48000 |   695 | D02    |
|   107 | Vergese  |  15000 |  NULL | D01    |
|   108 | Nachaobi |  29000 |  NULL | D05    |
|   109 | Daribha  |  42000 |  NULL | D04    |
|   110 | Tanya    |  50000 |   467 | D05    |
+-------+----------+--------+-------+--------+
9 rows in set (0.00 sec)
```

Example 9.6 The following query selects the name and department number of all those employees who are earning salary between 20000 and 50000 (both values inclusive).

```
mysql> SELECT Ename, DeptId
    -> FROM EMPLOYEE
    -> WHERE Salary>=20000 AND Salary<=50000;
+----------+--------+
| Ename    | DeptId |
+----------+--------+
| Shabbir  | D01    |
| Joseph   | D03    |
| Sanya    | D02    |
| Nachaobi | D05    |
| Daribha  | D04    |
| Tanya    | D05    |
+----------+--------+
6 rows in set (0.00 sec)
SELECT  * FROM  EMPLOYEE
WHERE Salary > 5000 OR DeptId= 20;
```

The query in example 9.6 defines a range that can also be checked using a comparison operator BETWEEN, as shown below:

```
mysql> SELECT  Ename, DeptId
    -> FROM EMPLOYEE
    -> WHERE Salary BETWEEN 20000 AND 50000;
+----------+--------+
| Ename    | DeptId |
+----------+--------+
| Shabbir  | D01    |
| Joseph   | D03    |
| Sanya    | D02    |
| Nachaobi | D05    |
| Daribha  | D04    |
| Tanya    | D05    |
+----------+--------+
6 rows in set (0.03 sec)
```

Note: The BETWEEN operator defines the range of values in which the column value must fall into, to make the condition true.

Example 9.7 The following query selects details of all the employees who work in the departments having deptid D01, D02 or D04.

Think and Reflect

What will happen if in the above query we write "Aaliya" as "AALIYA" or "aaliya" or "AaLIYA"? Will the query generate the same output or an error?

Activity 9.8

Compare the output produced by the query in Example 9.6 and the output of the following query and differentiate between the OR and AND operators.

© NCERT not to be republished

Reprint 2026-27

```
mysql> SELECT  *
    -> FROM EMPLOYEE
    -> WHERE DeptId = 'D01' OR DeptId = 'D02' OR
DeptId = 'D04';
    +-------+----------+--------+-------+--------+
    | EmpNo | Ename    | Salary | Bonus | DeptId |
    +-------+----------+--------+-------+--------+
    |   101 | Aaliya   |  10000 |   234 | D02    |
    |   102 | Kritika  |  60000 |   123 | D01    |
    |   103 | Shabbir  |  45000 |   566 | D01    |
    |   104 | Gurpreet |  19000 |   565 | D04    |
    |   106 | Sanya    |  48000 |   695 | D02    |
    |   107 | Vergese  |  15000 |  NULL | D01    |
    |   109 | Daribha  |  42000 |  NULL | D04    |
    +-------+----------+--------+-------+--------+
    7 rows in set (0.00 sec)
```

(E) Membership operator IN

The IN operator compares a value with a set of values and returns true if the value belongs to that set. The above query can be rewritten using IN operator as shown below:

```
mysql> SELECT  * FROM EMPLOYEE
    -> WHERE DeptId IN ('D01', 'D02' , 'D04');
    +-------+----------+--------+-------+--------+
    | EmpNo | Ename    | Salary | Bonus | DeptId |
    +-------+----------+--------+-------+--------+
    |   101 | Aaliya   |  10000 |   234 | D02    |
    |   102 | Kritika  |  60000 |   123 | D01    |
    |   103 | Shabbir  |  45000 |   566 | D01    |
    |   104 | Gurpreet |  19000 |   565 | D04    |
    |   106 | Sanya    |  48000 |   695 | D02    |
    |   107 | Vergese  |  15000 |  NULL | D01    |
    |   109 | Daribha  |  42000 |  NULL | D04    |
    +-------+----------+--------+-------+--------+
    7 rows in set (0.00 sec)
```

Example 9.8 The following query selects details of all the employees except those working in department number D01 or D02.

```
mysql> SELECT * FROM EMPLOYEE
    -> WHERE DeptId NOT IN('D01', 'D02');
    +-------+----------+--------+-------+--------+
    | EmpNo | Ename    | Salary | Bonus | DeptId |
    +-------+----------+--------+-------+--------+
    |   104 | Gurpreet |  19000 |   565 | D04    |
    |   105 | Joseph   |  34000 |   875 | D03    |
    |   108 | Nachaobi |  29000 |  NULL | D05    |
    |   109 | Daribha  |  42000 |  NULL | D04    |
    |   110 | Tanya    |  50000 |   467 | D05    |
    +-------+----------+--------+-------+--------+
    5 rows in set (0.00 sec)
```

Note: Here we need to combine NOT with IN as we want to retrieve all records except with DeptId D01 and D02.

© NCERT not to be republished

(F) ORDER BY Clause

ORDER BY clause is used to display data in an ordered form with respect to a specified column. By default, ORDER BY displays records in ascending order of the specified column's values. To display the records in descending order, the DESC (means descending) keyword needs to be written with that column.

Example 9.9 The following query selects details of all the employees in ascending order of their salaries.

```
mysql> SELECT  * FROM EMPLOYEE
    -> ORDER BY Salary;
+-------+----------+--------+-------+--------+
| EmpNo | Ename    | Salary | Bonus | DeptId |
+-------+----------+--------+-------+--------+
|   101 | Aaliya   |  10000 |   234 | D02    |
|   107 | Vergese  |  15000 |  NULL | D01    |
|   104 | Gurpreet |  19000 |   565 | D04    |
|   108 | Nachaobi |  29000 |  NULL | D05    |
|   105 | Joseph   |  34000 |   875 | D03    |
|   109 | Daribha  |  42000 |  NULL | D04    |
|   103 | Shabbir  |  45000 |   566 | D01    |
|   106 | Sanya    |  48000 |   695 | D02    |
|   110 | Tanya    |  50000 |   467 | D05    |
|   102 | Kritika  |  60000 |   123 | D01    |
+-------+----------+--------+-------+--------+
10 rows in set (0.05 sec)
```

Example 9.10 Select details of all the employees in descending order of their salaries.

```
mysql> SELECT  * FROM EMPLOYEE
    -> ORDER BY Salary DESC;
+-------+----------+--------+-------+--------+
| EmpNo | Ename    | Salary | Bonus | DeptId |
+-------+----------+--------+-------+--------+
|   102 | Kritika  |  60000 |   123 | D01    |
|   110 | Tanya    |  50000 |   467 | D05    |
|   106 | Sanya    |  48000 |   695 | D02    |
|   103 | Shabbir  |  45000 |   566 | D01    |
|   109 | Daribha  |  42000 |  NULL | D04    |
|   105 | Joseph   |  34000 |   875 | D03    |
|   108 | Nachaobi |  29000 |  NULL | D05    |
|   104 | Gurpreet |  19000 |   565 | D04    |
|   107 | Vergese  |  15000 |  NULL | D01    |
|   101 | Aaliya   |  10000 |   234 | D02    |
+-------+----------+--------+-------+--------+
10 rows in set (0.00 sec)
```

Activity 9.9

Execute the following 2 queries and find out what will happen if we specify two columns in the ORDER BY clause:

```
SELECT  * FROM
EMPLOYEE
ORDER BY Salary,
Bonus;

SELECT  * FROM
EMPLOYEE
ORDER BY
Salary,Bonus
DESC;
```

(G) Handling NULL Values

SQL supports a special value called NULL to represent a missing or unknown value. For example, the Gphone column in the GUARDIAN table can have missing value for certain records. Hence, NULL is used to represent such unknown values. It is important to note that NULL

© NCERT not to be republished

Reprint 2026-27

is different from 0 (zero). Also, any arithmetic operation performed with NULL value gives NULL. For example: 5 + NULL = NULL because NULL is unknown hence the result is also unknown. In order to check for NULL value in a column, we use IS NULL operator.

Example 9.11 The following query selects details of all those employees who have not been given a bonus. This implies that the bonus column will be blank.

```
mysql> SELECT * FROM EMPLOYEE
    -> WHERE Bonus IS NULL;
    +-------+----------+--------+-------+--------+
    | EmpNo | Ename    | Salary | Bonus | DeptId |
    +-------+----------+--------+-------+--------+
    |   107 | Vergese  |  15000 |  NULL | D01    |
    |   108 | Nachaobi |  29000 |  NULL | D05    |
    |   109 | Daribha  |  42000 |  NULL | D04    |
    +-------+----------+--------+-------+--------+
    3 rows in set (0.00 sec)
```

Example 9.12 The following query selects names of all employees who have been given a bonus (i.e., Bonus is not null) and works in the department D01.

```
mysql> SELECT EName FROM EMPLOYEE
    -> WHERE Bonus IS NOT NULL
    -> AND DeptID = 'D01';
    +----------+
    | EName    |
    +----------+
    | Kritika  |
    | Shabbir  |
    +----------+
    2 rows in set (0.00 sec)
```

(H) Substring pattern matching

Many a times we come across situations where we do not want to query by matching exact text or value. Rather, we are interested to find matching of only a few characters or values in column values. For example, to find out names starting with "T" or to find out pin codes starting with '60'. This is called substring pattern matching. We cannot match such patterns using = operator as we are not looking for an exact match. SQL provides a LIKE operator that can be used with the WHERE clause to search for a specified pattern in a column.

The LIKE operator makes use of the following two wild card characters:

- % (per cent)- used to represent zero, one, or multiple characters

© NCERT not to be republished

- _ (underscore)- used to represent exactly a single character

Example 9.13 The following query selects details of all those employees whose name starts with 'K'.

```
mysql> SELECT  * FROM EMPLOYEE
    -> WHERE Ename like 'K%';
    +-------+---------+--------+-------+--------+
    | EmpNo | Ename   | Salary | Bonus | DeptId |
    +-------+---------+--------+-------+--------+
    |   102 | Kritika |  60000 |   123 | D01    |
    +-------+---------+--------+-------+--------+
    1 row in set (0.00 sec)
```

Example 9.14 The following query selects details of all those employees whose name ends with 'a', and gets a salary more than 45000.

```
mysql> SELECT  * FROM EMPLOYEE
    -> WHERE Ename like '%a'
    -> AND Salary > 45000;

    +-------+---------+--------+-------+--------+
    | EmpNo | Ename   | Salary | Bonus | DeptId |
    +-------+---------+--------+-------+--------+
    |   102 | Kritika |  60000 |   123 | D01    |
    |   106 | Sanya   |  48000 |   695 | D02    |
    |   110 | Tanya   |  50000 |   467 | D05    |
    +-------+---------+--------+-------+--------+
    3 rows in set (0.00 sec)
```

Example 9.15 The following query selects details of all those employees whose name consists of exactly 5 letters and starts with any letter but has 'ANYA' after that.

```
mysql> SELECT  * FROM EMPLOYEE
    -> WHERE Ename like '_ANYA';
    +-------+-------+--------+-------+--------+
    | EmpNo | Ename | Salary | Bonus | DeptId |
    +-------+-------+--------+-------+--------+
    |   106 | Sanya |  48000 |   695 | D02    |
    |   110 | Tanya |  50000 |   467 | D05    |
    +-------+-------+--------+-------+--------+
    2 rows in set (0.00 sec)
```

Example 9.16 The following query selects names of all employees containing 'se' as a substring in name.

```
mysql> SELECT  Ename FROM EMPLOYEE
    -> WHERE Ename like '%se%';
    +---------+
    | Ename   |
    +---------+
    | Joseph  |
    | Vergese |
    +---------+
    2 rows in set (0.00 sec)
```

Think and Reflect

When we type first letter of a contact name in our contact list in our mobile phones all the names containing that character are displayed. Can you relate SQL statement with the process? List other real-life situations where you can visualise a SQL statement in operation.

© NCERT not to be republished

Reprint 2026-27

Example 9.17 The following query selects names of all employees containing 'a' as the second character.

```
mysql> SELECT  EName FROM EMPLOYEE
    -> WHERE Ename like '_a%';
    +----------+
    | EName    |
    +----------+
    | Aaliya   |
    | Sanya    |
    | Nachaobi |
    | Daribha  |
    | Tanya    |
    +----------+
    5 rows in set (0.00 sec)
```

9.7 Data Updation and Deletion

Updation and deletion of data are also part of SQL Data Manipulation Language (DML). In this section, we are going to apply these two data manipulation methods on the StudentAttendance database given in section 9.4.

9.7.1 Data Updation

We may need to make changes in the value(s) of one or more columns of existing records in a table. For example, we may require some changes in address, phone number or spelling of name, etc. The UPDATE statement is used to make such modifications in existing data.

Syntax:

```
UPDATE table_name
SET attribute1 = value1, attribute2 = value2, ...
WHERE condition;
```

STUDENT Table 9.7 has NULL value in GUID for the student with roll number 3. Suppose students with roll numbers 3 and 5 are siblings. Then, in the STUDENT table, we need to fill the GUID value for the student with roll number 3 as 101010101010. In order to update or change GUID of a particular row (record), we need to specify that record using WHERE clause, as shown below:

```
mysql> UPDATE STUDENT
    -> SET GUID = 101010101010
    -> WHERE RollNumber = 3;
Query OK, 1 row affected (0.06 sec) Rows matched: 1
Changed: 1 Warnings: 0
```

We can then verify the updated data using the statement SELECT * FROM STUDENT.

© NCERT not to be republished

Caution : If we miss the where clause in the UPDATE statement then the GUID of all the records will be changed to 101010101010.

We can also update values for more than one column using the UPDATE statement. Suppose, the guardian with GUID 466444444666 has requested to change Address to 'WZ - 68, Azad Avenue, Bijnour, MP' and Phone number to '4817362092'.

```
mysql> UPDATE GUARDIAN
    -> SET GAddress = 'WZ - 68, Azad Avenue,
-> Bijnour, MP', GPhone = 9010810547
    -> WHERE  GUID = 466444444666;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM GUARDIAN ;
+------------+--------------+------------+-------------------------------------+
|GUID        |GName         |Gphone      |GAddress                             |
+------------+--------------+------------+-------------------------------------+
|444444444444|Amit Ahuja    |5711492685  |G-35, Ashok vihar, Delhi             |
|111111111111|Baichung Bhutia|3612967082 |Flat no. 5, Darjeeling Appt., Shimla |
|101010101010|Himanshu Shah |4726309212  |26/77, West Patel Nagar, Ahmedabad   |
|333333333333|Danny Dsouza  |NULL        |S -13, Ashok Village, Daman          |
|466444444666|Sujata P.     |3801923168  |WZ - 68, Azad Avenue, Bijnour, MP    |
+------------+--------------+------------+-------------------------------------+
5 rows in set (0.00 sec)
```

9.7.2 Data Deletion

DELETE statement is used to delete/remove one or more records from a table.

Syntax:

```
DELETE FROM table_name
WHERE condition;
```

Suppose the student with roll number 2 has left the school. We can use the following MySQL statement to delete that record from the STUDENT table.

```
mysql> DELETE FROM STUDENT WHERE RollNumber = 2;
Query OK, 1 row affected (0.06 sec)
```

```
mysql> SELECT * FROM STUDENT ;
+------------+--------------+--------------+--------------+
| RollNumber | SName        | SDateofBirth | GUID         |
+------------+--------------+--------------+--------------+
|          1 | Atharv Ahuja | 2003-05-15   | 444444444444 |
|          3 | Taleem Shah  | 2002-02-28   | 101010101010 |
|          4 | John Dsouza  | 2003-08-18   | 333333333333 |
|          5 | Ali Shah     | 2003-07-05   | 101010101010 |
|          6 | Manika P.    | 2002-03-10   | 466444444666 |
+------------+--------------+--------------+--------------+
5 rows in set (0.00 sec)
```

Caution: Like UPDATE statement, we need to be careful to include the WHERE clause while using a DELETE statement to delete records in a table. Otherwise, all the records in the table will get deleted.

© NCERT not to be republished

9.8 Functions in SQL

In this section, we will understand how to use single row functions, multiple row functions, group records based on some criteria, and working on multiple tables using SQL.

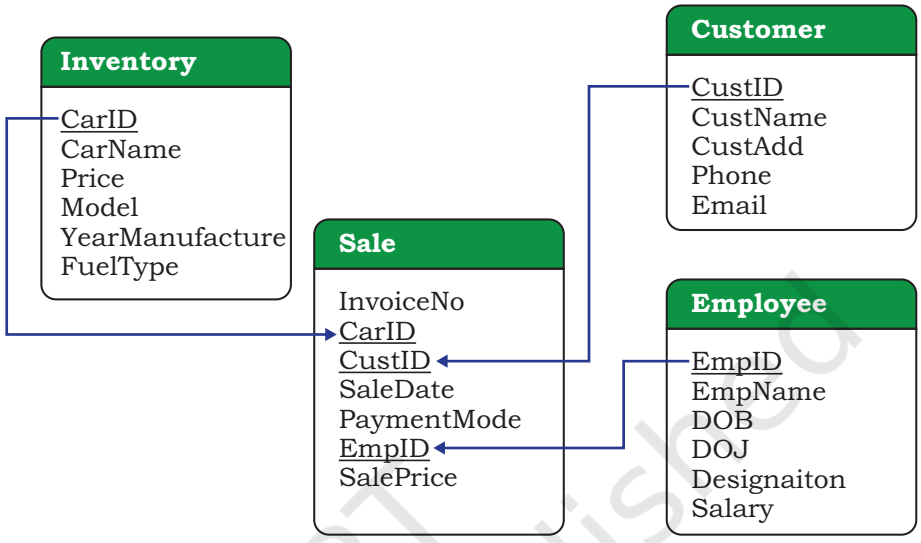


Figure 9.2: Schema diagram of database CARSHOWROOM

Let us create a database called CARSHOWROOM having the schema as shown in Figure 9.2 It has the following four relations:

1) INVENTORY: Stores name, price, model, year of manufacturing, and fuel type for each car in inventory of the showroom,
2) CUSTOMER: Stores customer id, name, address, phone number and email for each customer,
3) SALE: Stores the invoice number, car id, customer id, sale date, mode of payment, sales person's employee id and selling price of the car sold,
4) EMPLOYEE: Stores employee id, name, date of birth, date of joining, designation and salary of each employee in the showroom.

The records of the four relations are shown in Tables 9.9, 9.10, 9.11, and 9.12, respectively.

Table 9.9 INVENTORY

```
mysql> SELECT * FROM INVENTORY;
+-------+-------------+-----------+------------+------------------+-------------+
| CarId | CarName     | Price     | Model      | YearManufacture | Fueltype    |
+-------+-------------+-----------+------------+------------------+-------------+
| D001  | Dzire       | 582613.00 | LXI        |             2017 | Petrol      |
```

© NCERT not to be republished

```
| D002  | Dzire       | 673112.00 | VXI       |            2018 | Petrol      |
| B001  | Baleno      | 567031.00 | Sigma1.2  |            2019 | Petrol      |
| B002  | Baleno      | 647858.00 | Delta1.2  |            2018 | Petrol      |
| E001  | EECO        | 355205.00 | 5 STR STD |            2017 | CNG         |
| E002  | EECO        | 654914.00 | CARE      |            2018 | CNG         |
| S001  | SWIFT       | 514000.00 | LXI       |            2017 | Petrol      |
| S002  | SWIFT       | 614000.00 | VXI       |            2018 | Petrol      |
+-------+-------------+-----------+-----------+-----------------+-------------+
8 rows in set (0.00 sec)
```

Table 9.10 CUSTOMER

```
mysql> SELECT * FROM CUSTOMER;
+--------+-------------+-------------------------+------------+---------------------+
| CustId | CustName    | CustAdd                 | Phone      | Email               |
+--------+-------------+-------------------------+------------+---------------------+
| C0001  | Amit Saha   | L-10, Pitampura         | 4564587852 | amitsaha2@gmail.com |
| C0002  | Rehnuma     | J-12, SAKET             | 5527688761 | rehnuma@hotmail.com |
| C0003  | Charvi Nayyar| 10/9, FF, Rohini       | 6811635425 | charvi123@yahoo.com |
| C0004  | Gurpreet    | A-10/2, SF, Mayur Vihar | 3511056125 | gur_singh@yahoo.com |
+--------+-------------+-------------------------+------------+---------------------+
4 rows in set (0.00 sec)
```

Table 9.11 SALE

```
mysql> SELECT * FROM SALE;
+-----------+-------+--------+------------+-------------+-------+-----------+
| InvoiceNo | CarId | CustId | SaleDate   | PaymentMode | EmpID | SalePrice |
+-----------+-------+--------+------------+-------------+-------+-----------+
| I00001    | D001  | C0001  | 2019-01-24 | Credit Card | E004  | 613248.00 |
| I00002    | S001  | C0002  | 2018-12-12 | Online      | E001  | 590321.00 |
| I00003    | S002  | C0004  | 2019-01-25 | Cheque      | E010  | 604000.00 |
| I00004    | D002  | C0001  | 2018-10-15 | Bank Finance| E007  | 659982.00 |
| I00005    | E001  | C0003  | 2018-12-20 | Credit Card | E002  | 369310.00 |
| I00006    | S002  | C0002  | 2019-01-30 | Bank Finance| E007  | 620214.00 |
+-----------+-------+--------+------------+-------------+-------+-----------+
6 rows in set (0.00 sec)
```

Table 9.12 EMPLOYEE

```
mysql> SELECT * FROM EMPLOYEE;
+-------+----------+------------+------------+--------------+--------+
| EmpID | EmpName  | DOB        | DOJ        | Designation  | Salary |
+-------+----------+------------+------------+--------------+--------+
| E001  | Rushil   | 1994-07-10 | 2017-12-12 | Salesman     |  25550 |
| E002  | Sanjay   | 1990-03-12 | 2016-06-05 | Salesman     |  33100 |
| E003  | Zohar    | 1975-08-30 | 1999-01-08 | Peon         |  20000 |
| E004  | Arpit    | 1989-06-06 | 2010-12-02 | Salesman     |  39100 |
| E006  | Sanjucta | 1985-11-03 | 2012-07-01 | Receptionist |  27350 |
| E007  | Mayank   | 1993-04-03 | 2017-01-01 | Salesman     |  27352 |
| E010  | Rajkumar | 1987-02-26 | 2013-10-23 | Salesman     |  31111 |
+-------+----------+------------+------------+--------------+--------+
7 rows in set (0.00 sec)
```

We know that a function is used to perform some particular task and it returns zero or more values as a result. Functions are useful while writing SQL queries also. Functions can be applied to work on single or multiple records (rows) of a table. Depending on their application in one or multiple rows, SQL functions are

© NCERT not to be republished

categorised as Single Row functions and Aggregate functions.

9.8.1 Single Row Functions

These are also known as Scalar functions. Single row functions are applied on a single value and return a single value. Figure 9.3 lists different single row functions under three categories — Numeric (Math), String, Date and Time.

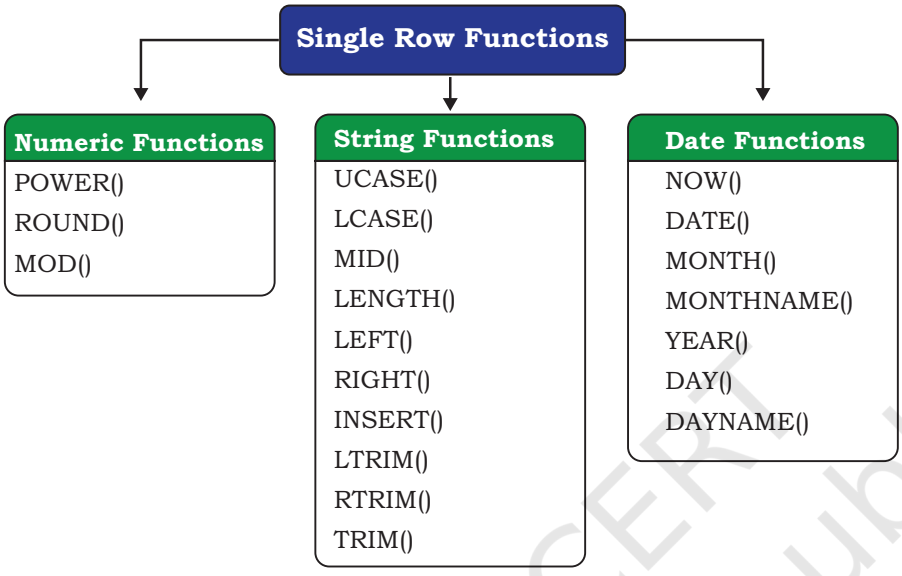


Figure 9.3: Three categories of single-row functions in SQL

Math Functions accept numeric value as input and return a numeric value as a result. String Functions accept character value as input and return either character or numeric values as output. Date and Time functions accept date and time value as input and return numeric or string or Date and Time as output.

(A) Math Functions

Three commonly used numeric functions are POWER(), ROUND() and MOD(). Their usage along with syntax is given in Table 9.13.

Table 9.13 Math Functions

Function	Description	Example with output

POWER(X,Y)
can also be written as POW(X,Y)
	Calculates X to the power Y.	`mysql> SELECT POWER(2,3);`
		`Output:`
		`8`

ROUND(N,D)	Rounds off number N to D number of decimal places.
	Note: If D=0, then it rounds off the number to the nearest integer.
		`mysql>SELECT ROUND(2912.564, 1);`
		`Output:`
		`2912.6`
		`mysql> SELECT ROUND(283.2);`
		`Output:`
		`283`

MOD(A, B)	Returns the remain der after dividing number A by number B.	`mysql> SELECT MOD(21, 2);`
		`Output:`
		`1`

© NCERT not to be republished

Example 9.18 In order to increase sales, suppose the car dealer decides to offer his customers to pay the total amount in 10 easy EMIs (equal monthly instalments). Assume that EMIs are required to be in multiples of 10000. For that, the dealer wants to list the CarID and Price along with the following data from the Inventory table:

a) Calculate GST as 12 per cent of Price and display the result after rounding it off to one decimal place.

```
mysql> SELECT ROUND(12/100*Price,1) "GST" FROM
INVENTORY;
+------------+
| GST        |
+------------+
| 69913.6    |
| 80773.4    |
| 68043.7    |
| 77743.0    |
| 42624.6    |
| 78589.7    |
| 61680.0    |
| 73680.0    |
+------------+
8 rows in set (0.00 sec)
```

b) Add a new column FinalPrice to the table inventory which will have the value as sum of Price and 12 per cent of the GST.

```
mysql> ALTER TABLE INVENTORY ADD(FinalPrice
Numeric(10,1));
Query OK, 8 rows affected (0.03 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> UPDATE INVENTORY SET
FinalPrice=Price+Round(Price*12/100,1);
Query OK, 8 rows affected (0.01 sec)
Rows matched: 8  Changed: 8  Warnings: 0
```

```
mysql> SELECT * FROM INVENTORY;
+-------+--------+----------+----------+----------------+----------+------------+
| CarId |CarName | Price    | Model    |YearManufacture | FuelType | FinalPrice |
+-------+--------+----------+----------+----------------+----------+------------+
| D001  | Dzire  | 582613.00 | LXI      | 2017           | Petrol   | 652526.6   |
| D002  | Dzire  | 673112.00 | VXI      | 2018           | Petrol   | 753885.4   |
| B001  | Baleno | 567031.00 | Sigma1.2 | 2019           | Petrol   | 635074.7   |
| B002  | Baleno | 647858.00 | Delta1.2 | 2018           | Petrol   | 725601.0   |
| E001  | EECO   | 355205.00 | 5 STR STD | 2017           | CNG      | 397829.6   |
| E002  | EECO   | 654914.00 | CARE     | 2018           | CNG      | 733503.7   |
| S001  | SWIFT  | 514000.00 | LXI      | 2017           | Petrol   | 575680.0   |
| S002  | SWIFT  | 614000.00 | VXI      | 2018           | Petrol   | 687680.0   |
+-------+--------+----------+----------+----------------+----------+------------+
8 rows in set (0.00 sec)
```

c) Calculate and display the amount to be paid each month (in multiples of 1000) which is to be calculated after dividing the FinalPrice of the car into 10 instalments.

Activity 9.10

Using the table SALE of CARSHOWROOM database, write SQL queries for the following:

a) Display the InvoiceNo and commission value rounded off to zero decimal places.

b) Display the details of SALE where payment mode is credit card.

© NCERT not to be republished

Reprint 2026-27

d) After dividing the amount into EMIs, find out the remaining amount to be paid immediately, by performing modular division.

Following SQL query can be used to solve the above mentioned (c) and (d) problem:

```
mysql> SELECT CarId, FinalPrice, ROUND(FinalPrice-
MOD(FinalPrice,1000)/10,0) "EMI",
MOD(FinalPrice,10000) "Remaining Amount" FROM
INVENTORY;
+-------+-------------+---------+------------------+
| CarId | FinalPrice  | EMI     | Remaining Amount |
+-------+-------------+---------+------------------+
| D001  |   652526.6  | 652474  | 2526.6           |
| D002  |   753885.4  | 753797  | 3885.4           |
| B001  |   635074.7  | 635067  | 5074.7           |
| B002  |   725601.0  | 725541  | 5601.0           |
| E001  |   397829.6  | 397747  | 7829.6           |
| E002  |   733503.7  | 733453  | 3503.7           |
| S001  |   575680.0  | 575612  | 5680.0           |
| S002  |   687680.0  | 687612  | 7680.0           |
+-------+-------------+---------+------------------+
8 rows in set (0.00 sec)
```

Example 9.19

a) Let us now add a new column Commission to the SALE table. The column Commission should have a total length of 7 in which 2 decimal places to be there.

```
mysql> ALTER TABLE SALE ADD(Commission
Numeric(7,2));
Query OK, 6 rows affected (0.34 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

b) Let us now calculate commission for sales agents as 12% of the SalePrice, Insert the values to the newly added column Commission and then display records of the table SALE where commission > 73000.

```
mysql> UPDATE SALE SET
Commission=12/100*SalePrice;
Query OK, 6 rows affected (0.06 sec)
Rows matched: 6  Changed: 6  Warnings: 0
```

```
mysql> SELECT * FROM SALE WHERE Commission > 73000;
+----------+------+--------+-----------+-------------+-------+-----------+------------+
|invoiceno |carid | custid | saledate  | paymentmode | empid | saleprice | Commission |
+----------+------+--------+-----------+-------------+-------+-----------+------------+
| I00001   |D001 | C0001  | 2019-01-24 | Credit Card | E004  | 613248.00 | 73589.64   |
| I00004   |D002 | C0001  | 2018-10-15 | Bank Finance| E007  | 659982.00 | 79198.84   |
| I00006   |S002 | C0002  | 2019-01-30 | Bank Finance | E007 | 620214.00 | 74425.68   |
+----------+------+--------+-----------+-------------+-------+-----------+------------+
3 rows in set (0.02 sec)
```

© NCERT not to be republished

c) Display InvoiceNo, SalePrice and Commission such that commission value is rounded off to 0.

```
mysql> SELECT InvoiceNo, SalePrice,
Round(Commission,0) FROM SALE;
+-----------+-----------+---------------------+
| InvoiceNo | SalePrice | Round(Commission,0) |
+-----------+-----------+---------------------+
| I00001    | 613248.00 |                73590 |
| I00002    | 590321.00 |                70839 |
| I00003    | 604000.00 |                72480 |
| I00004    | 659982.00 |                79198 |
| I00005    | 369310.00 |                44317 |
| I00006    | 620214.00 |                74426 |
+-----------+-----------+---------------------+
6 rows in set (0.00 sec)
```

Activity 9.11

Using the table INVENTORY from CARSHOWROOM database, write sql queries for the following:

a) Convert the CarMake to uppercase if its value starts with the letter 'B'.

b) If the length of the car's model is greater than 4 then fetch the substring starting from position 3 till the end from attribute Model.

(B) String Functions

String functions can perform various operations on alphanumeric data which are stored in a table. They can be used to change the case (uppercase to lowercase or vice-versa), extract a substring, calculate the length of a string and so on. String functions and their usage are shown in Table 9.14.

Table 9.14 String Functions

Function	Description	Example with output
UCASE(string) OR UPPER(string)	converts string into uppercase.	mysql> SELECT UCASE("Informatics Practices"); Output: INFORMATICS PRACTICES
LOWER(string) OR LCASE(string)	converts string into lowercase.	mysql> SELECT LOWER("Informatics Practices"); Output: informatics practices

MID(string, pos, n)
OR
SUBSTRING(string, pos, n)
OR
SUBSTR(string, pos, n)
	Returns a substring of size n starting from the specified position (pos) of the string. If n is not specified, it returns the substring from the position pos till end of the string.	mysql> SELECT MID("Informatics", 3, 4);
		Output:
		form
		
		mysql> SELECT MID('Informatics',7);
		Output:
		atics

LENGTH(string)	Return the number of characters in the specified string.	mysql> SELECT LENGTH("Informatics"); Output: 11
LEFT(string, N)	Returns N number of characters from the left side of the string.	mysql> SELECT LEFT("Computer", 4); Output: Comp

© NCERT not to be republished

Reprint 2026-27

RIGHT(string, N)	Returns N number of characters from the right side of the string.	mysql> SELECT RIGHT("SCIENCE", 3); NCE
INSTR(string, substring)	Returns the position of the first occurrence of the substring in the given string. Returns 0, if the substring is not present in the string.	mysql> SELECT INSTR("Informatics", "ma"); Output: 6
LTRIM(string)	Returns the given string after removing leading white space characters.	mysql> SELECT LENGTH(" DELHI"), LENGTH(LTRIM(" DELHI")); Output: +--------+--------+ | 7 | 5 | +--------+--------+ 1 row in set (0.00 sec)
RTRIM(string)	Returns the given string after removing trailing white space characters.	mysql>SELECT LENGTH("PEN ")LENGTH(RTRIM("PEN ")); Output: +--------+--------+ | 5 | 3 | +--------+--------+ 1 row in set (0.00 sec)
TRIM(string)	Returns the given string after removing both leading and trailing white space characters.	mysql> SELECT LENGTH(" MADAM "),LENGTH(TRIM(" MADAM ")); Output: +--------+--------+ | 9 | 5 | +--------+--------+ 1 row in set (0.00 sec)

Activity 9.12

Using the table EMPLOYEE from CARSHOWROOM database, write SQL queries for the following:

a) Display employee name and the last 2 characters of his EmpId.

b) Display designation of employee and the position of character 'e' in designation, if present.

Example 9.20 Let us use Customer relation shown in Table 9.10 to understand the working of string functions.

a) Display customer name in lower case and customer email in upper case from table CUSTOMER.

```
mysql> SELECT LOWER(CustName), UPPER(Email) FROM
CUSTOMER;
+-----------------+---------------------+
| LOWER(CustName) | UPPER(Email)        |
+-----------------+---------------------+
| amit saha       | AMITSAHA2@GMAIL.COM |
| rehnuma         | REHNUMA@HOTMAIL.COM |
| charvi nayyar   | CHARVI123@YAHOO.COM |
| gurpreet        | GUR_SINGH@YAHOO.COM |
+-----------------+---------------------+
4 rows in set (0.00 sec)
```

b) Display the length of the email and part of the email from the email id before the character '@'. Note - Do not print '@'.

```
mysql> SELECT LENGTH(Email), LEFT(Email, INSTR(Email,
"@")-1) FROM CUSTOMER;
```

© NCERT not to be republished

```
+---------------+----------------------------------+
| LENGTH(Email) | LEFT(Email, INSTR(Email, "@")-1) |
+---------------+----------------------------------+
|            19 | amitsaha2                        |
|            19 | rehnuma                          |
|            19 | charvi123                        |
|            19 | gur_singh                        |
+---------------+----------------------------------+
4 rows in set (0.03 sec)
```

The function INSTR will return the position of "@" in the email address. So, to print email id without "@" we have to use position -1.

c) Let us assume that four-digit area code is reflected in the mobile number starting from position number 3. For example, 1851 is the area code of mobile number 9818511338. Now, write the SQL query to display the area code of the customer living in Rohini.

```
mysql> SELECT MID(Phone,3,4) FROM CUSTOMER WHERE
CustAdd like '%Rohini%';
 +----------------+
 | MID(Phone,3,4) |
 +----------------+
 | 1163           |
 +----------------+
 1 row in set (0.00 sec)
```

d) Display emails after removing the domain name extension ".com" from emails of the customers.

```
mysql> SELECT TRIM(".com" from Email) FROM
CUSTOMER;
+--------------------------+
| TRIM(".com" FROM Email) |
+--------------------------+
| amitsaha2@gmail          |
| rehnuma@hotmail          |
| charvi123@yahoo          |
| gur_singh@yahoo          |
+--------------------------+
4    rows in set (0.00 sec)
```

e) Display details of all the customers having yahoo emails only.

```
mysql> SELECT * FROM CUSTOMER WHERE Email LIKE "%yahoo%";
+--------+-------------+-------------------------+------------+----------------------+
| CustID | CustName    | CustAdd                 | Phone      | Email                |
+--------+-------------+-------------------------+------------+----------------------+
| C0003  | Charvi Nayyar| 10/9, FF, Rohini        | 6811635425 | charvi123@yahoo.com  |
| C0004  | Gurpreet    | A-10/2, SF, Mayur Vihar | 3511056125 | gur_singh@yahoo.com  |
+--------+-------------+-------------------------+------------+----------------------+
2 rows in set (0.00 sec)
```

Activity 9.13

Using the table EMPLOYEE of CARSHOWROOM database, list the day of birth for all employees whose salary is more than 25000.

Think and Reflect

Can we use arithmetic operators (+, -. *, or /) on date functions?

(C) Date and Time Functions

There are various functions that are used to perform operations on date and time data. Some of the operations

© NCERT not to be republished

Reprint 2026-27

include displaying the current date, extracting each element of a date (day, month and year), displaying day of the week and so on. Table 9.15 explains various date and time functions.

Table 9.15 Date Functions

Function	Description	Example with output

NOW()	It returns the current system date and time.	`mysql> SELECT NOW();`
		`Output:`
		`2019-07-11 19:41:17`

DATE()	It returns the date part from the given date/time expression.	`mysql> SELECT DATE(NOW());`
		`Output:`
		`2019-07-11`

MONTH(date)	It returns the month in numeric form from the date.	`mysql> SELECT MONTH(NOW());`
		`Output:`
		`7`

MONTHNAME(date)	It returns the month name from the specified date.	`mysql> SELECT MONTHNAME("2003-11-28");`
		`Output:`
		`November`

YEAR(date)	It returns the year from the date.	`mysql> SELECT YEAR("2003-10-03");`
		`Output:`
		`2003`

DAY(date)	It returns the day part from the date.	`mysql> SELECT DAY("2003-03-24");`
		`Output:`
		`24`

DAYNAME(date)	It returns the name of the day from the date.	`mysql> SELECT DAYNAME("2019-07-11");`
		`Output:`
		`Thursday`

Example 9.21 Let us use the EMPLOYEE table of CARSHOWROOM database to illustrate the working of some of the date and time functions.

a) Select the day, month number and year of joining of all employees.

```
mysql> SELECT DAY(DOJ), MONTH(DOJ), YEAR(DOJ)
FROM EMPLOYEE;
+----------+------------+-----------+
| DAY(DOJ) | MONTH(DOJ) | YEAR(DOJ) |
+----------+------------+-----------+
|       12 |         12 |      2017 |
|        5 |          6 |      2016 |
|        8 |          1 |      1999 |
|        2 |         12 |      2010 |
|        1 |          7 |      2012 |
|        1 |          1 |      2017 |
|       23 |         10 |      2013 |
+----------+------------+-----------+
7 rows in set (0.03 sec)
```

Activity 9.14

a) Find sum of Sale Price of the cars purchased by the customer having ID C0001 from table SALE.

b) Find the maximum and minimum commission from the SALE table.

© NCERT not to be republished

b) If the date of joining is not a Sunday, then display it in the following format "Wednesday, 26, November, 1979."

```
mysql> SELECT DAYNAME(DOJ), DAY(DOJ),
MONTHNAME(DOJ), YEAR(DOJ) FROM EMPLOYEE WHERE
DAYNAME(DOJ)!='Sunday';
+--------------+----------+----------------+-----------+
| DAYNAME(DOJ) | DAY(DOJ) | MONTHNAME(DOJ) | YEAR(DOJ) |
+--------------+----------+----------------+-----------+
| Tuesday      |       12 | December       |      2017 |
| Friday       |        8 | January        |      1999 |
| Thursday     |        2 | December       |      2010 |
| Wednesday    |       23 | October        |      2013 |
+--------------+----------+----------------+-----------+
 4 rows in set (0.00 sec)
```

9.8.2 Aggregate Functions

Aggregate functions are also called Multiple Row functions. These functions work on a set of records as a whole and return a single value for each column of the records on which the function is applied. Table 9.16 shows the differences between single row functions and multiple row functions. Table 9.17 describes some of the aggregate functions along with their usage. Note that column must be of numeric type.

Table 9.16 Differences between Single and Multiple Row Functions

Single Row Function	Multiple row function
1. It operates on a single row at a time. 2. It returns one result per row. 3. It can be used in Select, Where, and Order by clause. 4. Math, String and Date functions are examples of single row functions.	1. It operates on groups of rows. 2. It returns one result for a group of rows. 3. It can be used in the select clause only. 4. Max(), Min(), Avg(), Sum(), Count() and Count(*) are examples of multiple row functions.

Table 9.17 Aggregate Functions in SQL

Function	Description	Example with output

MAX(column)	Returns the largest value from the specified column.	`mysql> SELECT MAX(Price) FROM INVENTORY;`
		`Output:`
		`673112.00`

MIN(column)	Returns the smallest value from the specified column.	`mysql> SELECT MIN(Price) FROM INVENTORY;`
		`Output:`
		`355205.00`

© NCERT not to be republished

Reprint 2026-27

AVG(column)	Returns the average of the values in the specified column.	mysql> SELECT AVG(Price) FROM INVENTORY; Output: 576091.625000
SUM(column)	Returns the sum of the values for the specified column.	mysql> SELECT SUM(Price) FROM INVENTORY; Output: 4608733.00
COUNT(*)	Returns the number of records in a table. Note: In order to display the number of records that matches a particular criteria in the table, we have to use COUNT(*) with WHERE clause.	mysql> SELECT COUNT(*) from MANAGER; +----------+ | count(*) | +----------+ | 4 | +----------+ 1 row in set (0.00 sec)
COUNT(column)	Returns the number of values in the specified column ignoring the NULL values. Note: In this example, let us consider a MANAGER table having two attributes and four records.	mysql> SELECT * from MANAGER; +------ +---------+ | MNO | MEMNAME | +------ +---------+ | 1 | AMIT | | 2 | KAVREET | | 3 | KAVITA | | 4 | NULL | +------+----------+ 4 rows in set (0.00 sec) mysql> SELECT COUNT(MEMNAME) FROM MANAGER; +----------------+ | COUNT(MEMNAME) | +----------------+ | 3 | +----------------+ 1 row in set (0.01 sec)

Example 9.22

a) Display the total number of records from table INVENTORY having a model as VXI.

```
mysql> SELECT COUNT(*) FROM INVENTORY WHERE
Model="VXI";
+----------+
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)
```

b) Display the total number of different types of Models available from table INVENTORY.

```
mysql> SELECT COUNT(DISTINCT Model) FROM
INVENTORY;
```

© NCERT not to be republished

```
+-----------------------+
| COUNT(DISTINCT MODEL) |
+-----------------------+
|                     6 |
+-----------------------+
1 row in set (0.09 sec)
```

c) Display the average price of all the cars with Model LXI from table INVENTORY.

```
mysql> SELECT AVG(Price) FROM INVENTORY WHERE
Model="LXI";
+---------------+
| AVG(Price)    |
+---------------+
| 548306.500000 |
+---------------+
1 row in set (0.03 sec)
```

Activity 9.15

a) List the total number of cars sold by each employee.

b) List the maximum sale made by each employee.

9.9 GROUP BY Clause in SQL

At times we need to fetch a group of rows on the basis of common values in a column. This can be done using a group by clause. It groups the rows together that contains the same values in a specified column. We can use the aggregate functions (COUNT, MAX, MIN, AVG and SUM) to work on the grouped values. HAVING Clause in SQL is used to specify conditions on the rows with Group By clause.

Consider the SALE table from the CARSHOWROOM database:

```
mysql> SELECT * FROM SALE;
+---------+------+-------+------------+-------------+------+----------+----------+
|InvoiceNo|CarId |CustId | SaleDate   | PaymentMode |EmpID |SalePrice |Commission|
+---------+------+-------+------------+-------------+------+----------+----------+
| I00001  | D001 | C0001 | 2019-01-24 | Credit Card | E004 |613248.00 | 73589.64 |
| I00002  | S001 | C0002 | 2018-12-12 | Online      | E001 |590321.00 | 70838.52 |
| I00003  | S002 | C0004 | 2019-01-25 | Cheque      | E010 |604000.00 | 72480.00 |
| I00004  | D002 | C0001 | 2018-10-15 | Bank Finance| E007 |659982.00 | 79198.84 |
| I00005  | E001 | C0003 | 2018-12-20 | Credit Card | E002 |369310.00 | 44318.20 |
| I00006  | S002 | C0002 | 2019-01-30 | Bank Finance| E007 |620214.00 | 74425.68 |
+---------+------+-------+------------+-------------+------+----------+----------+
6 rows in set (0.11 sec)
```

CarID, CustID, SaleDate, PaymentMode, EmpID, SalePrice are the columns that can have rows with the same values in it. So, Group by clause can be used in these columns to find the number of records of a particular type (column), or to calculate the sum of the price of each car type.

© NCERT not to be republished

Reprint 2026-27

Example 9.23

a) Display the number of Cars purchased by each Customer from SALE table.

```
mysql> SELECT CustID, COUNT(*) "Number of Cars"
FROM SALE GROUP BY CustID;
+--------+---------------+
| CustID | Number of Cars |
+--------+---------------+
| C0001  |             2 |
| C0002  |             2 |
| C0003  |             1 |
| C0004  |             1 |
+--------+---------------+
4 rows in set (0.00 sec)
```

b) Display the Customer Id and number of cars purchased if the customer purchased more than 1 car from SALE table.

```
mysql> SELECT CustID, COUNT(*) FROM SALE GROUP BY
CustID HAVING Count(*)>1;
+--------+----------+
| CustID | COUNT(*) |
+--------+----------+
| C0001  |        2 |
| C0002  |        2 |
+--------+----------+
2 rows in set (0.30 sec)
```

c) Display the number of people in each category of payment mode from the table SALE.

```
mysql> SELECT PaymentMode, COUNT(PaymentMode)
FROM SALE GROUP BY Paymentmode ORDER BY
Paymentmode;
+-------------+--------------------+
| PaymentMode  | Count(PaymentMode) |
+-------------+--------------------+
| Bank Finance |                  2 |
| Cheque       |                  1 |
| Credit Card  |                  2 |
| Online       |                  1 |
+-------------+--------------------+
4 rows in set (0.00 sec)
```

d) Display the PaymentMode and number of payments made using that mode more than once.

```
mysql> SELECT PaymentMode, Count(PaymentMode)
FROM SALE GROUP BY Paymentmode HAVING COUNT(*)>1
ORDER BY Paymentmode;
+-------------+--------------------+
| PaymentMode  | Count(PaymentMode) |
+-------------+--------------------+
| Bank Finance |                  2 |
| Credit Card  |                  2 |
+-------------+--------------------+
2 rows in set (0.00 sec)
```

© NCERT not to be republished

9.10 Operations on Relations

We can perform certain operations on relations like Union, Intersection and Set Difference to merge the tuples of two tables. These three operations are binary operations as they work upon two tables. Note here that these operations can only be applied if both the relations have the same number of attributes and corresponding attributes in both tables have the same domain.

9.10.1 UNION (∪)

This operation is used to combine the selected rows of two tables at a time. If some rows are same in both the tables, then result of the Union operation will show those rows only once. Figure 9.4 shows union of two sets.

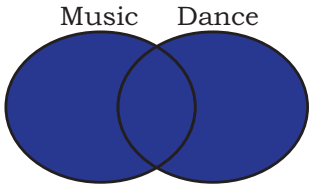


Figure 9.4: Union of two sets

Let us consider two relations DANCE and MUSIC shown in Tables 9.18 and 9.19 respectively.

Table 9.18 DANCE

SNo	Name	Class
1	Aastha	7A
2	Mahira	6A
3	Mohit	7B
4	Sanjay	7A

Table 9.19 MUSIC

SNo	Name	Class
1	Mehak	8A
2	Mahira	6A
3	Lavanya	7A
4	Sanjay	7A
5	Abhay	8A

If we need the list of students participating in either of events, then we have to apply UNION operation (represented by symbol U) on relations DANCE and MUSIC. The output of UNION operation is shown in Table 9.20.

© NCERT not to be republished

Reprint 2026-27

Table 9.20 DANCE ∪ MUSIC

SNo	Name	Class
1	Aastha	7A
2	Mahira	6A
3	Mohit	7B
4	Sanjay	7A
1	Mehak	8A
3	Lavanya	7A
5	Abhay	8A

9.10.2 INTERSECT (∩)

Intersect operation is used to get the common tuples from two tables and is represented by symbol ∩. Figure 9.5 shows intersection of two sets.

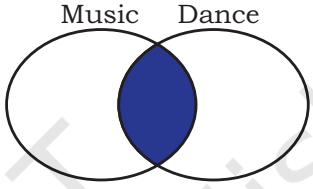


Figure 9.5: Intersection of two sets

Suppose, we have to display the list of students who are participating in both the events (DANCE and MUSIC), then intersection operation is to be applied on these two tables. The output of INTERSECT operation is shown in Table 9.21.

Table 9.21 DANCE ∩ MUSIC

SNo	Name	Class
2	Mahira	6A
4	Sanjay	7A

9.10.3 MINUS (-)

This operation is used to get tuples/rows which are in the first table but not in the second table and the operation is represented by the symbol - (minus). Figure 9.6 shows minus operation (also called set difference) between two sets.

© NCERT not to be republished

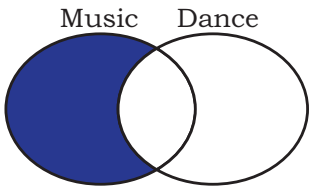


Figure 9.6: Difference of two sets

Suppose we want the list of students who are only participating in MUSIC and not in DANCE event. Then, we will use the MINUS operation, whose output is given in Table 9.22.

Table 9.22 DANCE - MUSIC

SNo	Name	Class
1	Mehak	8A
3	Lavanya	7A
5	Abhay	8A

9.10.4 Cartesian Product (X)

Cartesian product operation combines tuples from two relations. It results in all pairs of rows from the two input relations, regardless of whether or not they have the same values on common attributes. It is denoted as 'X'.

The degree of the resulting relation is calculated as the sum of the degrees of both the relations under consideration. The cardinality of the resulting relation is calculated as the product of the cardinality of relations on which cartesian product is applied. Let us use the relations DANCE and MUSIC to show the output of cartesian product. Note that both relations are of degree 3. The cardinality of relations DANCE and MUSIC is 4 and 5 respectively. Applying cartesian product on these two relations will result in a relation of degree 6 and cardinality 20, as shown in Table 9.23.

Table 9.23 DANCE X MUSIC

SNo	Name	Class	SNo	Name	Class
1	Aastha	7A	1	Mehak	8A
2	Mahira	6A	1	Mehak	8A
3	Mohit	7B	1	Mehak	8A
4	Sanjay	7A	1	Mehak	8A
1	Aastha	7A	2	Mahira	6A
2	Mahira	6A	2	Mahira	6A

© NCERT not to be republished

```
| 3 | Mohit  | 7B  |     2 | Mahira  | 6A  |
| 4 | Sanjay | 7A  |     2 | Mahira  | 6A  |
| 1 | Aastha | 7A  |     3 | Lavanya | 7A  |
| 2 | Mahira | 6A  |     3 | Lavanya | 7A  |
| 3 | Mohit  | 7B  |     3 | Lavanya | 7A  |
| 4 | Sanjay | 7A  |     3 | Lavanya | 7A  |
| 1 | Aastha | 7A  |     4 | Sanjay  | 7A  |
| 2 | Mahira | 6A  |     4 | Sanjay  | 7A  |
| 3 | Mohit  | 7B  |     4 | Sanjay  | 7A  |
| 4 | Sanjay | 7A  |     4 | Sanjay  | 7A  |
| 1 | Aastha | 7A  |     5 | Abhay   | 8A  |
| 2 | Mahira | 6A  |     5 | Abhay   | 8A  |
| 3 | Mohit  | 7B  |     5 | Abhay   | 8A  |
| 4 | Sanjay | 7A  |     5 | Abhay   | 8A  |
+---+--------+-----+-------+---------+-----+
20 rows in set (0.03 sec)
```

9.11 Using Two Relations in a Query

Till now we have written queries in SQL using a single relation only. In this section, we will learn to write queries using two relations.

9.11.1 Cartesian product on two tables

From the previous section, we learnt that application of operator cartesian product on two tables results in a table having all combinations of tuples from the underlying tables. When more than one table is to be used in a query, then we must specify the table names by separating commas in the FROM clause, as shown in Example 9.24. On execution of such a query, the DBMS (MySql) will first apply cartesian product on specified tables to have a single table. The following query of example 9.24 applies cartesian product on the two tables DANCE and MUSIC:

Example 9.24

a) Display all possible combinations of tuples of relations DANCE and MUSIC

```
mysql> SELECT * FROM DANCE, MUSIC;
```

As we are using SELECT * in the query, the output will be the Table 9.23 having degree 6 and cardinality 20.

b) From the all possible combinations of tuples of relations DANCE and MUSIC display only those rows such that the attribute name in both have the same value.

```
mysql> SELECT * FROM DANCE D, MUSIC M WHERE
D.Name = M.Name;
```

© NCERT not to be republished

Table 9.24 Tuples with same name

```
+------+--------+-------+------+--------+-------+
| Sno  | Name   | Class | Sno  | Name   | class |
+------+--------+-------+------+--------+-------+
|    2 | Mahira | 6A    |    2 | Mahira | 6A    |
|    4 | Sanjay | 7A    |    4 | Sanjay | 7A    |
+------+--------+-------+------+--------+-------+
2 rows in set (0.00 sec)
```

Note that in this query we have used table aliases (D for DANCE and M for MUSIC), just like column aliases (see Section 9.6.2) to refer to tables by shortened names. It is important to note that table alias is valid only for current query and the original table name cannot be used in the query if its alias is given in FROM clause.

9.11.2 JOIN on two tables

JOIN operation combines tuples from two tables on specified conditions. This is unlike cartesian product which make all possible combinations of tuples. While using the JOIN clause of SQL, we specify conditions on the related attributes of two tables within the FROM clause. Usually, such an attribute is the primary key in one table and foreign key in another table. Let us create two tables UNIFORM (UCode, UName, UColor) and COST (UCode, Size, Price) in the SchoolUniform database. UCode is Primary Key in table UNIFORM. UCode and Size is the Composite Key in table COST. Therefore, Ucode is a common attribute between the two tables which can be used to fetch the common data from both tables. Hence, we need to define Ucode as foreign key in the Price table while creating this table.

Table 9.25 Uniform table

```
+-------+-------+--------+
| Ucode | Uname | Ucolor |
+-------+-------+--------+
|   1   | Shirt | White  |
|   2   | Pant  | Grey   |
|   3   | Tie   | Blue   |
+-------+-------+--------+
```

Table 9.26 Cost table

```
+-------+------+-------+
| Ucode | Size | Price |
+-------+------+-------+
|   1   | L    |   580 |
|   1   | M    |   500 |
|   2   | L    |   890 |
|   2   | M    |   810 |
+-------+------+-------+
```

© NCERT not to be republished

Reprint 2026-27

Example 9.25 List the UCode, UName, UColor, Size and Price of related tuples of tables UNIFORM and COST.

The given query may be written in three different ways as given below.

a) Using condition in where clause

```
mysql> SELECT * FROM UNIFORM U, COST C WHERE
U.UCode = C.UCode;
```

Table 9.27 Output of the query

```
+-------+-------+--------+-------+-----+-------+
| UCode | UName | UColor | Ucode |Size | Price |
+-------+-------+--------+-------+-----+-------+
|   1   | Shirt | White  | 1     | L   |   580 |
|   1   | Shirt | White  | 1     | M   |   500 |
|   2   | Pant  | Grey   | 2     | L   |   890 |
|   2   | Pant  | Grey   | 2     | M   |   810 |
+-------+-------+--------+-------+-----+-------+
4 rows in set (0.08 sec)
```

As the attribute Ucode is in both tables, we need to use table alias to remove ambiguity. Hence, we have used qualifier with attribute UCode in SELECT and FROM clauses to indicate its scope.

b) Explicit use of JOIN clause

```
mysql> SELECT * FROM UNIFORM U JOIN COST C ON
U.Ucode=C.Ucode;
```

The output of the query is same as shown in Table 9.26. In this query we have used JOIN clause explicitly along with condition in From clause. Hence no condition needs to be given in where clause.

c) Explicit use of NATURAL JOIN clause

The output of queries (a) and (b) shown in Table 9.26 has a repetitive column Ucode having exactly the same values. This redundant column provides no additional information. There is an extension of JOIN operation called NATURAL JOIN which works similar to JOIN clause in SQL but removes the redundant attribute. This operator can be used to join the contents of two tables if there is one common attribute in both the tables. The above SQL query using NATURAL JOIN is shown below:

```
mysql> SELECT * FROM UNIFORM  NATURAL JOIN COST;
+-------+-------+--------+------+-------+
| UCode | UName | UColor | Size | Price |
+-------+-------+--------+------+-------+
|   1   | Shirt | White  | L    |  580  |
|   1   | Shirt | White  | M    |  500  |
|   2   | Pant  | Grey   | L    |   890 |
```

© NCERT not to be republished

```
|   2   | Pant  | Grey   | M    |   810 |
+-------+-------+--------+------+-------+
4 rows in set (0.17 sec)
```

It is clear from the output that the result of this query is same as that of queries written in (a) and (b) except that the attribute Ucode appears only once.

Following are some of the points to be considered while applying JOIN operations on two or more relations:

- If two tables are to be joined on equality condition on the common attribute, then one may use JOIN with ON clause or NATURAL JOIN in FROM clause. If three tables are to be joined on equality condition, then two JOIN or NATURAL JOIN are required.
- In general, N-1 joins are needed to combine N tables on equality condition.
- With JOIN clause, we may use any relational operators to combine tuples of two tables.

SUMMARY

- Database is a collection of related tables. MySQL is a 'relational' DBMS.
- DDL (Data Definition Language) includes SQL statements such as, Create table, Alter table and Drop table.
- DML (Data Manipulation Language) includes SQL statements such as, insert, select, update and delete.
- A table is a collection of rows and columns, where each row is a record and columns describe the feature of records.
- ALTER TABLE statement is used to make changes in the structure of a table like adding, removing or changing datatype of column(s).
- UPDATE statement is used to modify existing data in a table.
- WHERE clause in SQL query is used to enforce condition(s).
- DISTINCT clause is used to eliminate repetition and display the values only once.

© NCERT not to be republished

Reprint 2026-27

- The BETWEEN operator defines the range of values inclusive of boundary values.
- The IN operator selects values that match any value in the given list of values.
- NULL values can be tested using IS NULL and IS NOT NULL.
- ORDER BY clause is used to display the result of a SQL query in ascending or descending order with respect to specified attribute values. By default, the order is ascending.
- LIKE operator is used for pattern matching. % and _ are two wild card characters. The per cent (%) symbol is used to represent zero or more characters. The underscore (_) symbol is used to represent a single character.
- A Function is used to perform a particular task and return a value as a result.
- Single Row functions work on a single row of the table and return a single value.
- Multiple Row functions work on a set of records as a whole and return a single value. Examples include COUNT, MAX, MIN, AVG and SUM.
- GROUP BY function is used to group rows of a table that contain the same values in a specified column.
- Join is an operation which is used to combine rows from two or more tables based on one or more common fields between them.

EXERCISE

1. Answer the following questions:
   a) Define RDBMS. Name any two RDBMS software.
   b) What is the purpose of the following clauses in a select statement?
      i) ORDER BY
      ii) GROUP BY
   c) Site any two differences between Single Row Functions and Aggregate Functions.
   d) What do you understand by Cartesian Product?
   e) Differentiate between the following statements:

© NCERT not to be republished

Reprint 2026-27

i) ALTER and UPDATE
ii) DELETE and DROP

f) Write the name of the functions to perform the following operations:

i) To display the day like “Monday”, “Tuesday”, from the date when India got independence.

ii) To display the specified number of characters from a particular position of the given string.

iii) To display the name of the month in which you were born.

iv) To display your name in capital letters.

2. Write the output produced by the following SQL statements:

a) SELECT POW(2,3);

b) SELECT ROUND(342.9234,-1);

c) SELECT LENGTH("Informatics Practices");

d) SELECT YEAR(“1979/11/26”), MONTH(“1979/11/26”), DAY(“1979/11/26”), MONTHNAME(“1979/11/26”);

e) SELECT LEFT("INDIA",3), RIGHT("Computer Science",4), MID("Informatics",3,4), SUBSTR("Practices",3);

3. Consider the following MOVIE table and write the SQL queries based on it.

MovieID	MovieName	Category	ReleaseDate	ProductionCost	BusinessCost
001	Hindi_Movie	Musical	2018-04-23	124500	130000
002	Tamil_Movie	Action	2016-05-17	112000	118000
003	English_Movie	Horror	2017-08-06	245000	360000
004	Bengali_Movie	Adventure	2017-01-04	72000	100000
005	Telugu_Movie	Action	-	100000	-
006	Punjabi_Movie	Comedy	-	30500	-

a) Display all the information from the Movie table.

b) List business done by the movies showing only MovieID, MovieName and Total_Earning. Total_Earning to be calculated as the sum of ProductionCost and BusinessCost.

c) List the different categories of movies.

d) Find the net profit of each movie showing its MovieID, MovieName and NetProfit. Net Profit is to be calculated as the difference between Business Cost and Production Cost.

© NCERT not to be republished

Reprint 2026-27

e) List MovieID, MovieName and Cost for all movies with ProductionCost greater than 10,000 and less than 1,00,000.

f) List details of all movies which fall in the category of comedy or action.

g) List details of all movies which have not been released yet.

4. Suppose your school management has decided to conduct cricket matches between students of Class XI and Class XII. Students of each class are asked to join any one of the four teams – Team Titan, Team Rockers, Team Magnet and Team Hurricane. During summer vacations, various matches will be conducted between these teams. Help your sports teacher to do the following:
   a) Create a database "Sports".
   b) Create a table "TEAM" with following considerations:
      i) It should have a column TeamID for storing an integer value between 1 to 9, which refers to unique identification of a team.
      ii) Each TeamID should have its associated name (TeamName), which should be a string of length not less than 10 characters.
   c) Using table level constraint, make TeamID as the primary key.
   d) Show the structure of the table TEAM using a SQL statement.
   e) As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:
      ```
      Row 1: (1, Team Titan)
      Row 2: (2, Team Rockers)
      Row 3: (3, Team Magnet)
      Row 3: (4, Team Hurricane)
      ```
   f) Show the contents of the table TEAM using a DML statement.
   g) Now create another table MATCH_DETAILS and insert data as shown below. Choose appropriate data types and constraints for each attribute.

Table: MATCH_DETAILS

MatchID	MatchDate	FirstTeamID	SecondTeamID	FirstTeamScore	SecondTeamScore
M1	2018-07-17	1	2	90	86
M2	2018-07-18	3	4	45	48
M3	2018-07-19	1	3	78	56
M4	2018-07-19	2	4	56	67
M5	2018-07-18	1	4	32	87
M6	2018-07-17	2	3	67	51

5. Using the sports database containing two relations (TEAM, MATCH_DETAILS) and write the queries for the following:

© NCERT not to be republished

a) Display the MatchID of all those matches where both the teams have scored more than 70.
b) Display the MatchID of all those matches where FirstTeam has scored less than 70 but SecondTeam has scored more than 70.
c) Display the MatchID and date of matches played by Team 1 and won by it.
d) Display the MatchID of matches played by Team 2 and not won by it.
e) Change the name of the relation TEAM to T_DATA. Also change the attributes TeamID and TeamName to T_ID and T_NAME respectively.

6. A shop called Wonderful Garments who sells school uniforms maintains a database SCHOOLUNIFORM as shown below. It consisted of two relations - UNIFORM and COST. They made UniformCode as the primary key for UNIFORM relations. Further, they used UniformCode and Size to be composite keys for COSTrelation. By analysing the database schema and database state, specify SQL queries to rectify the following anomalies.

   a) M/S Wonderful Garments also keeps handkerchiefs of red colour, medium size of Rs. 100 each.

   b) INSERT INTO COST (UCode, Size, Price) values (7, 'M',100);

      When the above query is used to insert data, the values for the handkerchief without entering its details in the UNIFORM relation is entered. Make a provision so that the data can be entered in the COST table only if it is already there in the UNIFORM table.

   c) Further, they should be able to assign a new UCode to an item only if it has a valid UName. Write a query to add appropriate constraints to the SCHOOLUNIFORM database.

   d) Add the constraint so that the price of an item is always greater than zero.

7. Consider the following table named "Product", showing details of products being sold in a grocery shop.

PCode	PName	UPrice	Manufacturer
P01	Washing Powder	120	Surf
P02	Toothpaste	54	Colgate
P03	Soap	25	Lux
P04	Toothpaste	65	Pepsodent
P05	Soap	38	Dove
P06	Shampoo	245	Dove

© NCERT not to be republished

Reprint 2026-27

Write SQL queries for the following:

a) Create the table Product with appropriate data types and constraints.
b) Identify the primary key in Product.
c) List the Product Code, Product name and price in descending order of their product name. If PName is the same, then display the data in ascending order of price.
d) Add a new column Discount to the table Product.
e) Calculate the value of the discount in the table Product as 10 per cent of the UPrice for all those products where the UPrice is more than 100, otherwise the discount will be 0.
f) Increase the price by 12 per cent for all the products manufactured by Dove.
g) Display the total number of products manufactured by each manufacturer.

Write the output(s) produced by executing the following queries on the basis of the information given above in the table Product:

h) SELECT PName, avg(UPrice) FROM Product GROUP BY Pname;
i) SELECT DISTINCT Manufacturer FROM Product;
j) SELECT COUNT (DISTINCT PName) FROM Product;
k) SELECT PName, MAX(UPrice), MIN(UPrice) FROM Product GROUP BY PName;

8. Using the CARSHOWROOM database given in the chapter, write the SQL queries for the following:

a) Add a new column Discount in the INVENTORY table.
b) Set appropriate discount values for all cars keeping in mind the following:

(i) No discount is available on the LXI model.

(ii) VXI model gives a 10 per cent discount.

(iii) A 12 per cent discount is given on cars other than LXI model and VXI model.

c) Display the name of the costliest car with fuel type "Petrol".
d) Calculate the average discount and total discount available on Baleno cars.
e) List the total number of cars having no discount.

© NCERT not to be republished

Chapter 10

Computer Networks

12130CH10

"Hoaxes use weaknesses in human behavior to ensure they are replicated and distributed. In other words, hoaxes prey on the Human Operating System."

— Stewart Kirkpatrick

In this Chapter

- *Introduction to Computer Networks*
- *Evolution of Networking*
- *Types of Networks*
- *Network Devices*
- *Networking Topologies*
- *Identifying Nodes in a Networked Communication*
- *Internet, Web and the Internet of Things*
- *Domain Name System*

10.1 Introduction to Computer Networks

We are living in a connected world. Information is being produced, exchanged, and traced across the globe in real time. It's possible as almost everyone and everything in the digital world is interconnected through one way or the other.

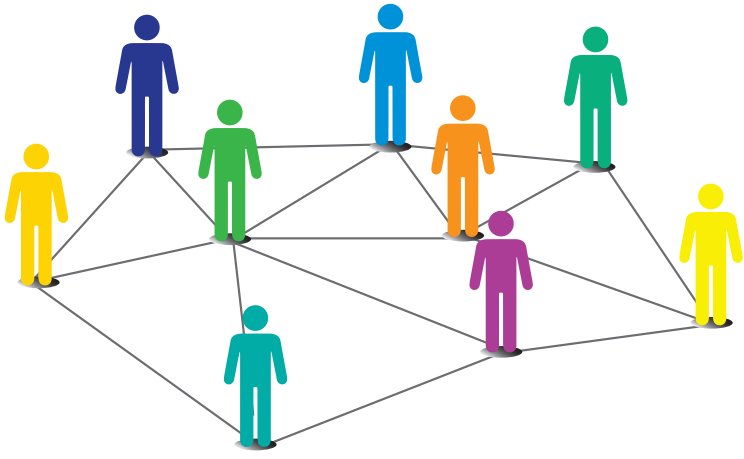


Figure 10.1: Interconnection forming a social network

© NCERT not to be republished

Activity 10.1

Identify some other networks in the real world.

A group of two or more similar things or people interconnected with each other is called network (Figure 10.1). Some of the examples of network in our everyday life includes:

- Social network
- Mobile network
- Network of computers
- Airlines, railway, banks, hospitals networks

A computer network (Figure 10.2) is an interconnection among two or more computers or computing devices. Such interconnection allows computers to share data and resources among each other. A basic network may connect a few computers placed in a room.

The network size may vary from small to large depending on the number of computers it connects. A computer network can include different types of hosts (also called nodes) like server, desktop, laptop, cellular phones.

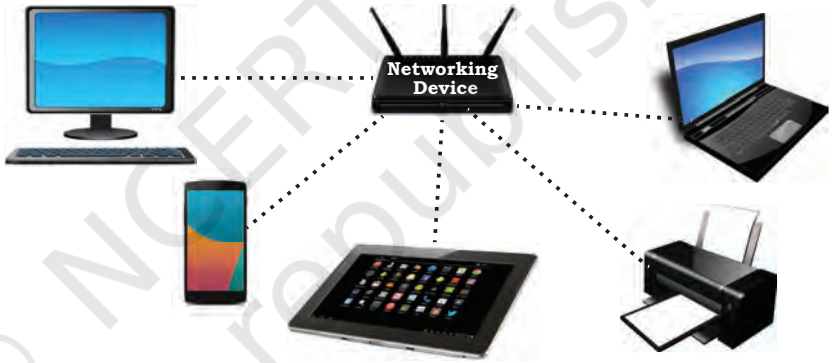


Figure 10.2: A computer network

Apart from computers, networks include networking devices like switch, router, modem, etc. Networking devices are used to connect multiple computers in different settings. For communication, data in a network is divided into smaller chunks called packets. These packets are then carried over a network. Devices in a network can be connected either through wired media like cables or wireless media like air.

In a communication network, each device that is a part of a network and that can receive, create, store or send data to different network routes is called a node. In the context of data communication, a node can be a device such as a modem, hub, bridge, switch, router, digital telephone handset, a printer, a computer or a server.

© NCERT not to be republished

Reprint 2026-27

Interconnectivity of computing devices in a network allows us to exchange information simultaneously with many parties through email, websites, audio/video calls, etc. Network allows sharing of resources. For example, a printer can be made available to multiple computers through a network; a networked storage can be accessed by multiple computers. People often connect their devices through hotspot, thus forming a small personal network.

Activity 10.2

Create a hotspot using a smartphone and connect other devices to it.

10.2 Evolution of Networking

In the 1960s a research project was commissioned by Advanced Research Projects Agency Network (ARPANET) in the U.S. Department of Defence to connect the academic and research institutions located at different places for scientific collaborations. The first message was communicated between the University of California, Los Angeles (UCLA) and Stanford Research Institute (SRI). Slowly but gradually, more and more organisations joined the ARPANET, and many independent smaller networks were formed. Few of the milestones in the magnificent journey of evolution of computer networks is depicted in the timeline shown in Figure 10.3.

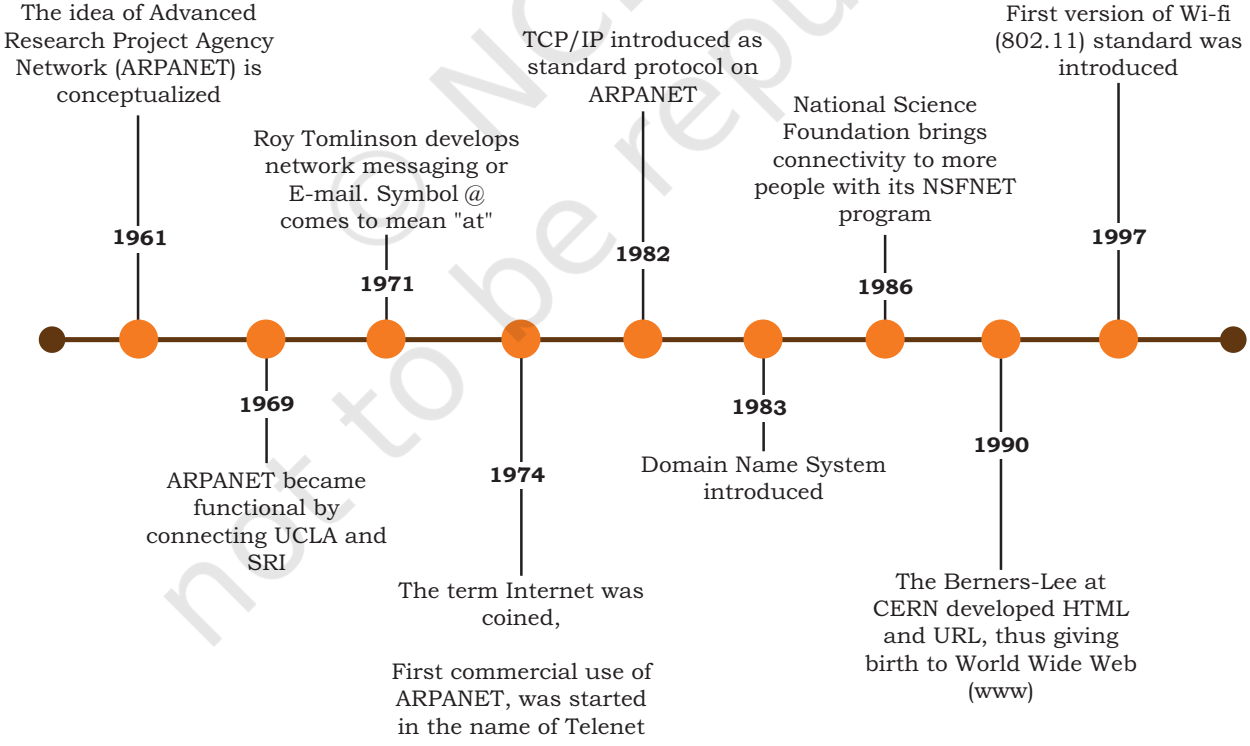


Figure 10.3: Timeline showing evolution of networking

© NCERT not to be republished

Reprint 2026-27

10.3 TYPES OF NETWORKS

There are various types of computer networks ranging from network of handheld devices (like mobile phones or tablets) connected through Wi-Fi or Bluetooth within a single room to the millions of computers spread across the globe. Some are connected wireless while others are connected through wires.

Based on the geographical area covered and data transfer rate, computer networks are broadly categorised as:

- PAN (Personal Area Network)
- LAN (Local Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

10.3.1 Personal Area Network (PAN)

It is a network formed by connecting a few personal devices like computers, laptops, mobile phones, smart phones, printers etc., as shown in Figure 10.4. All these devices lie within an approximate range of 10 metres. A personal area network may be wired or wireless. For example, a mobile phone connected to the laptop through USB forms a wired PAN while two smartphones communicating with each other through Bluetooth technology form a wireless PAN or WPAN.

Figure 10.4: A Personal Area Network

© NCERT not to be republished

Reprint 2026-27

10.3.2 Local Area Network (LAN)

It is a network that connects computers, mobile phones, tablet, mouse, printer, etc., placed at a limited distance. The geographical area covered by a LAN can range from a single room, a floor, an office having one or more buildings in the same premise, laboratory, a school, college, or university campus. The connectivity is done by means of wires, Ethernet cables, fibre optics, or Wi-Fi. A Local Area Network (LAN) is shown in Figure 10.5.

Think and Reflect

Explore and find out the minimum internet speed required to make a video call.

Figure 10.5: A Local Area Network

LAN is comparatively secure as only authentic users in the network can access other computers or shared resources. Users can print documents using a connected printer, upload/download documents and software to and from the local server. Such LANs provide the short range communication with the high speed data transfer rates. These types of networks can be extended up to 1 km. Data transfer in LAN is quite high, and usually varies from 10 Mbps (called Ethernet) to 1000 Mbps (called Gigabit Ethernet), where Mbps stands for Megabits per second. Ethernet is a set of rules that decides how computers and other devices connect with each other through cables in a local area network or LAN.

10.3.3 Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN) is an extended form of LAN which covers a larger geographical area like a city or a town. Data transfer rate in MAN also ranges in Mbps,

© NCERT not to be republished

Reprint 2026-27

but it is considerably less as compared to LAN. Cable TV network or cable based broadband internet services are examples of MAN. This kind of network can be extended up to 30-40 km. Sometimes, many LANs are connected together to form MAN, as shown in Figure 10.6.

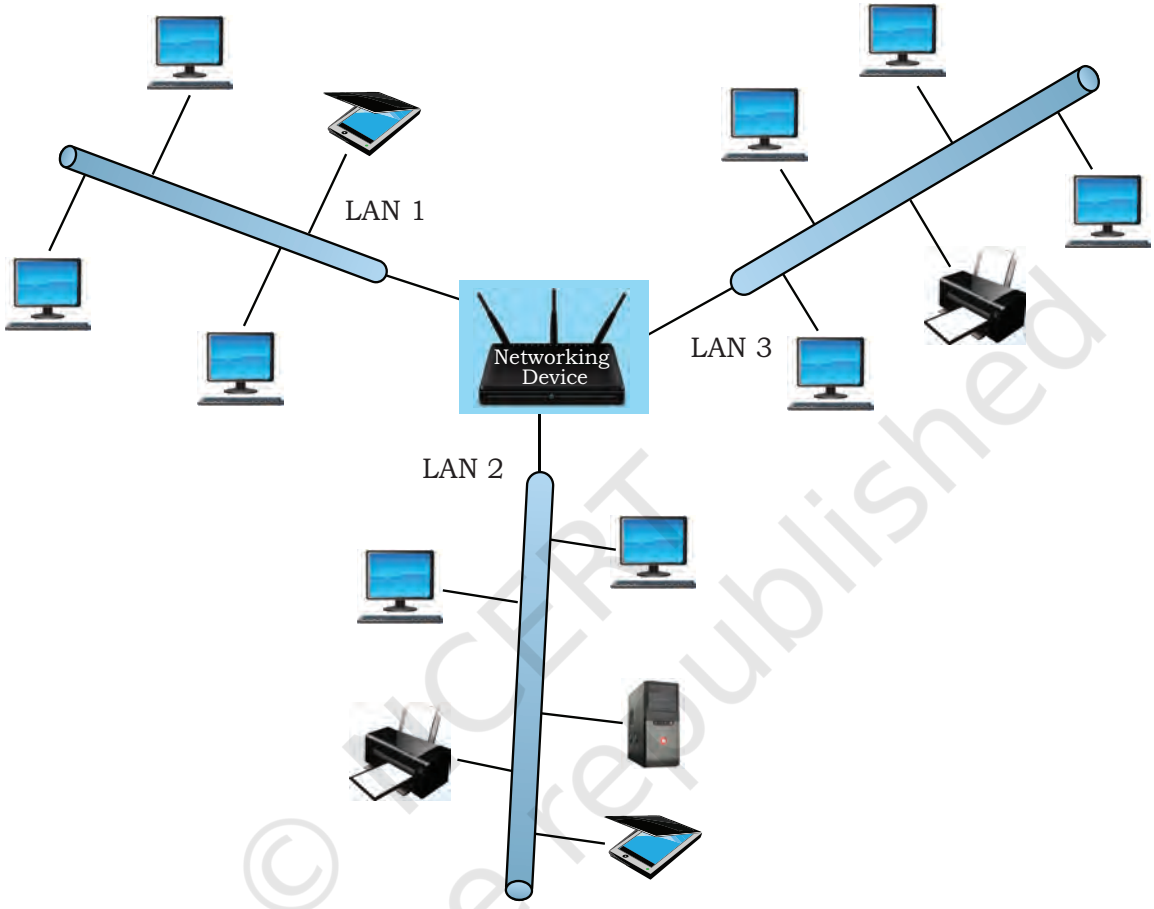


Figure 10.6: A Metropolitan Area Network

10.3.4 Wide Area Network (WAN)

Wide Area Network connects computers and other LANs and MANs, which are spread across different geographical locations of a country or in different countries or continents. A WAN could be formed by connecting a LAN to other LANs (Figure 10.7) via wired/wireless media. Large business, educational and government organisations connect their different branches in different locations across the world through WAN. The Internet is the largest WAN that connects billions of computers, smartphones and millions of LANs from different continents.

Think and Reflect

It is possible to access your bank account from any part of the world. Whether the bank's network is a LAN, MAN, WAN or any other type?

© NCERT not to be republished

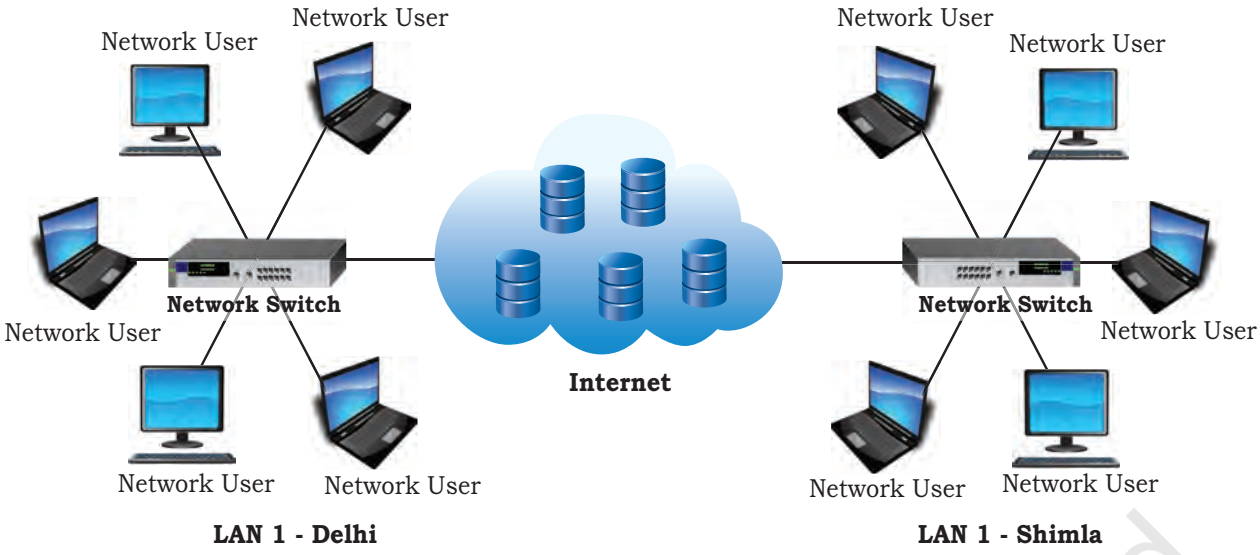


Figure 10.7: A Wide Area Network

10.4 Network Devices

To communicate data through different transmission media and to configure networks with different functionality, we require different devices like Modem, Hub, Switch, Repeater, Router, Gateway, etc. Let us explore them in detail.

10.4.1 Modem

Modem stands for 'MOdulator DEModulator'. It refers to a device used for conversion between analog signals and digital bits. We know computers store and process data in terms of 0s and 1s. However, to transmit data from a sender to a receiver, or while browsing the internet, digital data are converted to an analog signal and the medium (be it free-space or a physical media) carries the signal to the receiver. There are modems connected to both the source and destination nodes. The modem at the sender's end acts as a modulator that converts the digital data into analog signals. The modem at the receiver's end acts as a demodulator that converts the analog signals into digital data for the destination node to understand. Figure 10.8 shows connectivity using a modem.

© NCERT not to be republished

Reprint 2026-27

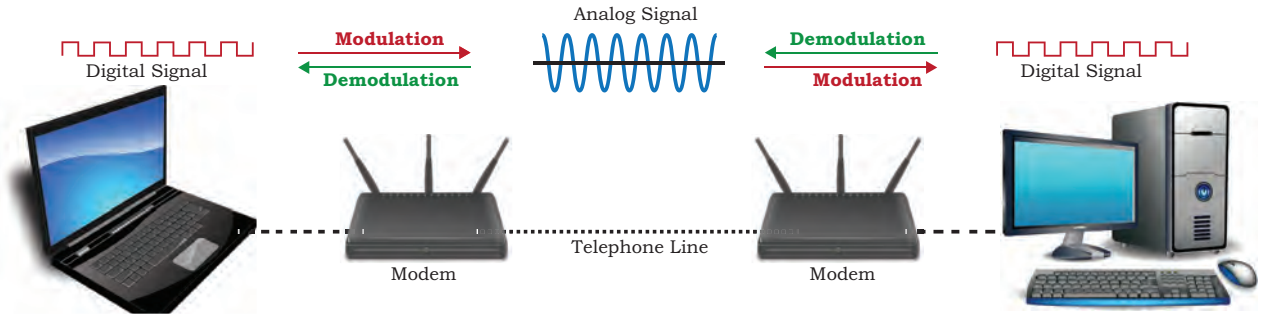


Figure 10.8: Use of modem

10.4.2 Ethernet Card

Ethernet card, also known as Network Interface Card (NIC card in short) is a network adapter used to set up a wired network. It acts as an interface between computer and the network. It is a circuit board mounted on the motherboard of a computer as shown in Figure 10.9. The Ethernet cable connects the computer to the network through NIC. Ethernet cards can support data transfer between 10 Mbps and 1 Gbps (1000 Mbps). Each NIC has a MAC address, which helps in uniquely identifying the computer on the network.

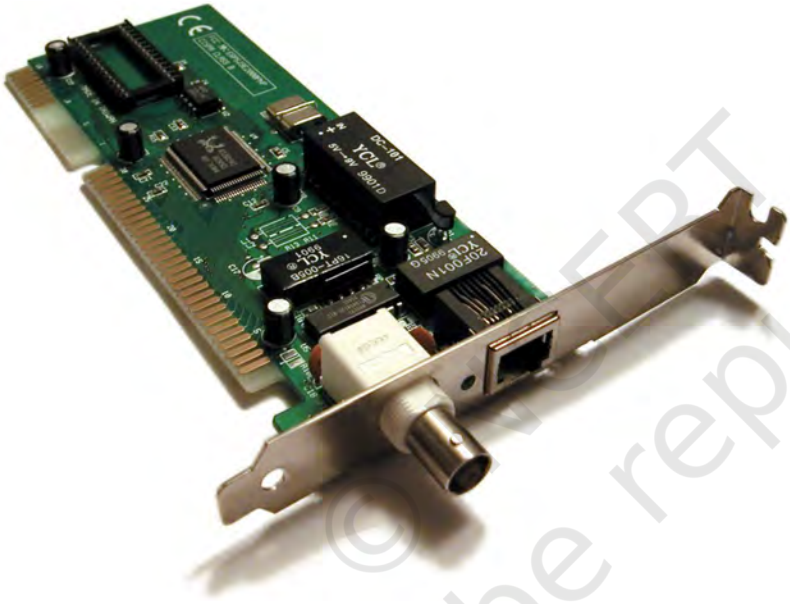


Figure 10.9: A Network Interface Card

Figure 10.10: RJ 45

10.4.3 RJ45

RJ 45 or Registered Jack-45 is an eight-pin connector (Figure 10.10) that is used exclusively with Ethernet cables for networking. It is a standard networking interface that can be seen at the end of all network cables. Basically, it is a small plastic plug that fits into RJ-45 jacks of the Ethernet cards present in various computing devices.

© NCERT not to be republished

10.4.4 Repeater

Data are carried in the form of signals over the cable. These signals can travel a specified distance (usually about 100 m). Signals lose their strength beyond this limit and become weak. In such conditions, original signals need to be regenerated.

A repeater is an analog device that works with signals on the cables to which it is connected. The weakened signal appearing on the cable is regenerated and put back on the cable by a repeater.

An Internet service provider (ISP) is any organisation that provides services for accessing the Internet.

10.4.5 Hub

An Ethernet hub (Figure 10.11) is a network device used to connect different devices through wires. Data arriving on any of the lines are sent out on all the others. The limitation of Hub is that if data from two devices come at the same time, they will collide.

Figure 10.11:A network hub with 8 ports

Activity 10.3

Find and list a few ISPs in your region.

10.4.6 Switch

A switch is a networking device (Figure 10.12) that plays a central role in a Local Area Network (LAN). Like a hub, a network switch is used to connect multiple computers or communicating devices. When data arrives, the switch extracts the destination address from the data packet and looks it up in a table to see where to send the packet. Thus, it sends signals to only selected devices instead of sending to all. It can forward multiple packets at the same time. A switch does not forward the signals which are noisy or corrupted. It drops such signals and asks the sender to resend it.

Figure 10.12:Cables connected to a network switch

Ethernet switches are common in homes/offices to connect multiple devices thus creating LANs or to access the Internet.

© NCERT not to be republished

10.4.7 Router

A router (Figure 10.13) is a network device that can receive the data, analyse it and transmit it to other networks. A router connects a local area network to the internet. Compared to a hub or a switch, a router has advanced capabilities as it can analyse the data being carried over a network, decide/alter how it is packaged, and send it to another network of a different type. For example, data has been divided into packets of a certain size. Suppose these packets are to be carried over a different type of network which cannot handle bigger packets. In such a case, the data is to be repackaged as smaller packets and then sent over the network by a router.

Figure 10.13:A router

A router can be wired or wireless. A wireless router can provide Wi-Fi access to smartphones and other devices. Usually, such routers also contain some ports to provide wired Internet access. These days, home Wi-Fi routers perform the dual task of a router and a modem/ switch. These routers connect to incoming broadband lines, from ISP (Internet Service Provider), and convert them to digital data for computing devices to process.

10.4.8 Gateway

As the term “Gateway” suggests, it is a key access point that acts as a “gate” between an organisation's network and the outside world of the Internet (Figure 10.14). Gateway serves as the entry and exit point of a network, as all data coming in or going out of a network must first pass through the gateway in order to use routing paths. Besides routing data packets, gateways also maintain information about the host network's internal connection paths and the identified paths of other remote networks. If a node from one network wants to communicate with a node of a foreign network, it will

© NCERT not to be republished

pass the data packet to the gateway, which then routes it to the destination using the best possible route.

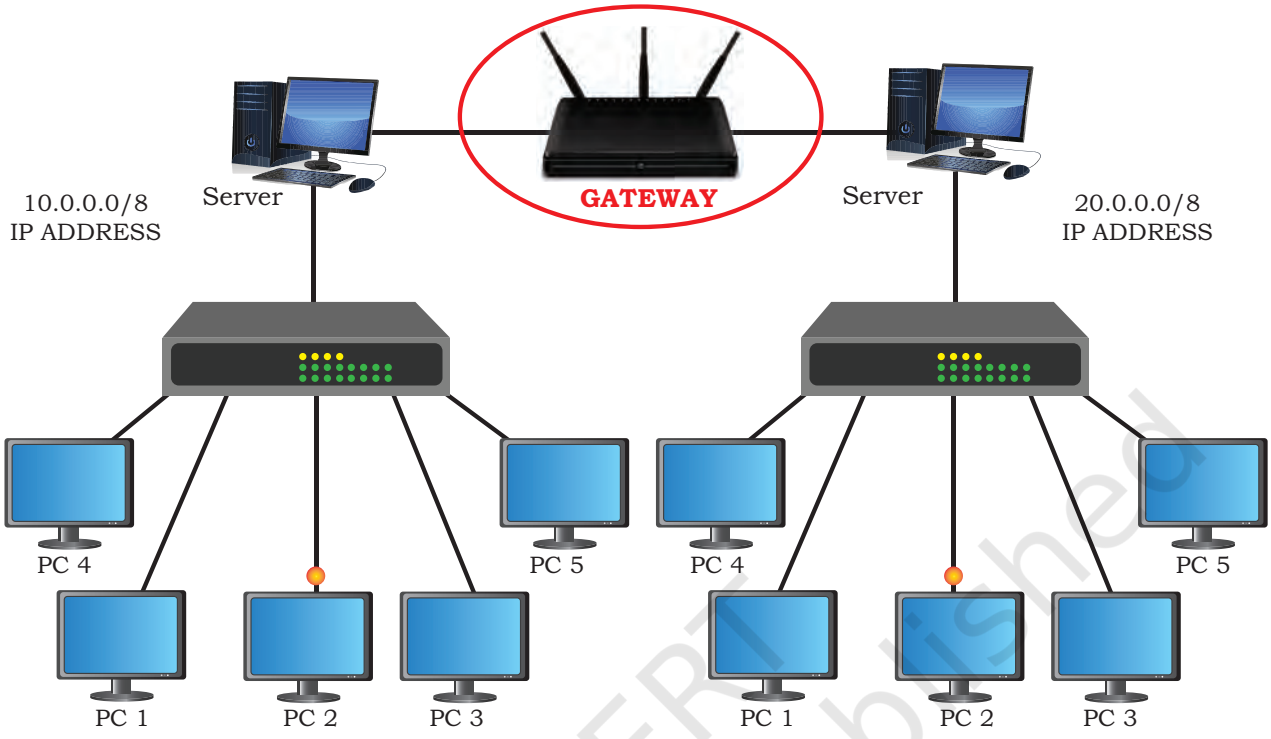


Figure 10.14:A network gateway

For simple Internet connectivity at homes, the gateway is usually the Internet Service Provider that provides access to the entire Internet. Generally, a router is configured to work as a gateway device in computer networks. But a gateway can be implemented completely in software, hardware, or a combination of both. Because a network gateway is placed at the edge of a network, the firewall is usually integrated with it.

10.5 Networking Topologies

We have already discussed that a number of computing devices are connected together to form a Local Area Network (LAN), and interconnections among millions of LANs forms the Internet. The arrangement of computers and other peripherals in a network is called its topology. Common network topologies are Mesh, Ring, Bus, Star and Tree.

© NCERT not to be republished

10.5.1 Mesh Topology

In this networking topology, each communicating device is connected with every other device in the network as shown in Figure 10.15. Such a network can handle large amounts of traffic since multiple nodes can transmit data simultaneously. Also, such networks are more reliable in the sense that even if a node gets down, it does not cause any break in the transmission of data between other nodes. This topology is also more secure as compared to other topologies because each cable between two nodes carries different data. However, wiring is complex and cabling cost is high in creating such networks and there are many redundant or unutilised connections.

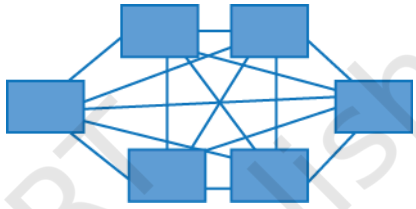


Figure 10.15:A mesh topology

To build a fully-connected mesh topology of n nodes, it requires n(n-1)/2 wires.

10.5.2 Ring Topology

In ring topology (Figure 10.16), each node is connected to two other devices, one each on either side, as shown in Figure 10.16. The nodes connected with each other thus forms a ring. The link in a ring topology is unidirectional. Thus, data can be transmitted in one direction only (clockwise or counterclockwise).

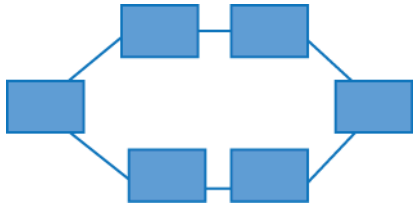


Figure 10.16:A ring topology

10.5.3 Bus Topology

In bus topology (Figure 10.17), each communicating device connects to a transmission medium, known as bus. Data sent from a node are passed on to the bus and hence are transmitted to the length of the bus in both directions. That means, data can be received by any of the nodes connected to the bus.

© NCERT not to be republished

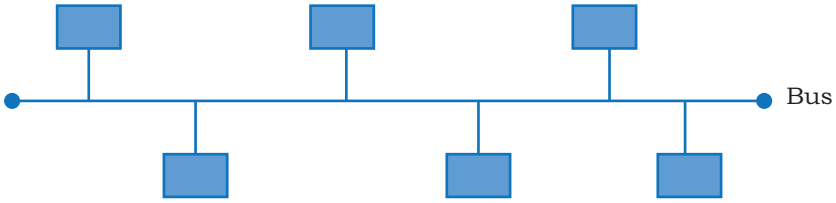


Figure 10.17:A bus topology

In this topology, a single backbone wire called bus is shared among the nodes, which makes it cheaper and easier to maintain. Both ring and bus topologies are considered to be less secure and less reliable.

10.5.4 Star Topology

In star topology (Figure 10.18), each communicating device is connected to a central node, which is a networking device like a hub or a switch, as shown in Figure 10.18.

Think and Reflect

How will a Bus and Ring topology behave in case a Node is down?

Star topology is considered very effective, efficient and fast as each device is directly connected with the central device. Although disturbance in one device will not affect the rest of the network, any failure in a central networking device may lead to the failure of complete network.

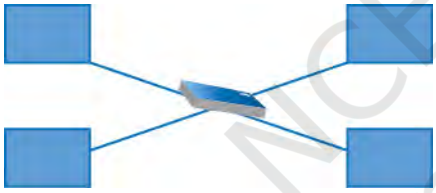


Figure 10.18:A star topology

The central node can be either a broadcasting device means data will be transmitted to all the nodes in the network, or a unicast device means the node can identify the destination and forward data to that node only.

10.5.5 Tree or Hybrid Topology

It is a hierarchical topology, in which there are multiple branches and each branch can have one or more basic topologies like star, ring and bus. Such topologies are usually realised in WANs where multiple LANs are connected. Those LANs may be in the form of a ring, bus or star. In figure 10.19, a hybrid topology is shown connecting 4-star topologies in a bus.

In this type of network, data transmitted from source first reaches the centralised device and from there the data passes through every branch where each branch can have links for more nodes.

© NCERT not to be republished

Reprint 2026-27

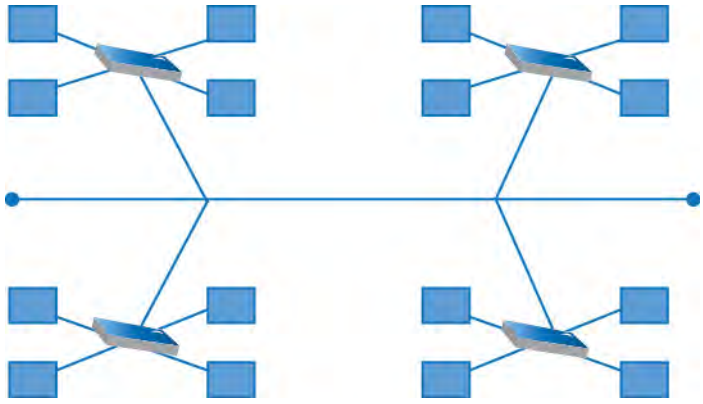


Figure 10.19:A hybrid topology

10.6 Identifying Nodes in a Networked Communication

Each node in a network should be uniquely identified so that a network device can identify the sender and receiver and decide a routing path to transmit data. Let us explore further and know how each node is distinguished in a network.

10.6.1 MAC Address

MAC stands for Media Access Control. The MAC address, also known as the physical or hardware address, is a unique value associated with a network adapter called a NIC. The MAC address is engraved on NIC at the time of manufacturing and thus it is a permanent address and cannot be changed under any circumstances. The machine on which the NIC is attached, can be physically identified on the network using its MAC address.

Each MAC address is a 12-digit hexadecimal numbers (48 bits in length), of which the first six digits (24 bits) contain the manufacturer's ID called Organisational Unique Identifier (OUI) and the later six digits (24 bits) represents the serial number assigned to the card by the manufacturer. A sample MAC address looks like:

FC:F8:AE:CE:7B:16

OUI — Unique Serial Number

Activity 10.4

Explore how can you find the MAC address of your computer system.

10.6.2 IP Address

IP address, also known as Internet Protocol address, is also a unique address that can be used to uniquely identify each node in a network. The IP addresses

© NCERT not to be republished

Reprint 2026-27

are assigned to each node in a network that uses the Internet Protocol for communication. Thus, if we know a computer's IP address, we can communicate with that computer from anywhere in the world. However, unlike MAC address, IP address can change if a node is removed from one network and connected to another network.

The initial IP Address called version 4 (IPV4 in short), is a 32 bit numeric address, written as four numbers separated by periods, where each number is the decimal (base-10) representation for an 8-bit binary (base-2) number and each can take any value from 0 - 255. A sample IPV4 address looks like:

```
192.168.0.178
```

With more and more devices getting connected to the Internet, it was realised that the 32-bit IP address will not be sufficient as it offers just under 4.3 billion unique addresses. Thus, a 128 bits IP address, called IP version 6 (IPV6 in short) was proposed. An IPv6 address is represented by eight groups of hexadecimal (base-16) numbers separated by colons. A sample IPV6 address looks like:

```
2001:CDBA:0000:0000:0000:0000:3257:9652
```

10.7 Internet, Web and the Internet of Things

The Internet is the global network of computing devices including desktop, laptop, servers, tablets, mobile phones, other handheld devices, printers, scanners, routers, switches, gateways, etc. Moreover, smart electronic appliances like TV, AC, refrigerator, fan, light, etc. can also communicate through a network. The list of such smart devices is always increasing e.g., drones, vehicles, door lock, security camera. We have already studied IoT and WoT in class 11.

The Internet is evolving every day and it is difficult to visualise or describe each and every aspect of the architecture of the Internet. Computers are either connected to a modem through a cable or wirelessly (Wi-Fi). That modem, be it wired or wireless, is connected to a local Internet Service Provider (ISP) who then connects to a national network. Many such ISPs connect together forming a regional network and regional networks connect together forming a national network, and such country-wise networks form the Internet backbone.

Think and Reflect

Do mobile phones have a MAC address? Is it different from the IMEI number of mobile phones?

© NCERT not to be republished

Think and Reflect

You are encouraged to take up any area of concern where you think IoT can be immensely beneficial and discuss it with your peers. An example for the same can be preventing road accidents.

The Internet today is a widespread network, and its influence is no longer limited to the technical fields of computer communications. It is being used by everyone in the society as is evident from the increasing use of online tools for education, creativity, entertainment, socialisation, and e-commerce.

10.7.1 The World Wide Web (WWW)

The World Wide Web (WWW) or web in short, is an ocean of information, stored in the form of trillions of interlinked web pages and web resources. The resources on the web can be shared or accessed through the Internet.

Earlier, to access files residing in different computers, one had to login individually to each computer through the Internet. Besides, files in different computers were sometimes in different formats, and it was difficult to understand each other's files and documents. Sir Tim Berners-Lee — a British computer scientist invented the revolutionary World Wide Web in 1990 by defining three fundamental technologies that lead to creation of web:

- HTML – HyperText Markup Language. It is a language which is used to design standardised Web Pages so that the Web contents can be read and understood from any computer. Basic structure of every webpage is designed using HTML.
- URI – Uniform Resource Identifier. It is a unique address or path for each resource located on the web. It is also known as Uniform Resource Locator (URL). Every page on the web has a unique URL. Examples are: https://www.mhrd.gov.in,http://www.ncert.nic.in,http://www.airindia.in, etc. URL is sometimes also called web address. However, a URL is not only the domain name. It contains other information that completes a web address, as depicted below:

- HTTP – The HyperText Transfer Protocol is a set of rules which is used to retrieve linked web pages across the web. The more secure and advanced version is HTTPS.

© NCERT not to be republished

Many people confuse the web with the Internet. The Internet as we know is the huge global network of interconnected computers, which may or may not have any file or webpage to share with the world. The web on the other hand is the interlinking of collection of Webpages on these computers which are accessible over the Internet. WWW today gives users access to a vast collection of information created and shared by people across the world. It is today the most popular information retrieval system

10.8 Domain Name System

The Internet is a vast ocean where information is available in the form of millions of websites. Each website is stored on a server which is connected to the Internet, which means each server has an IP address. Every device connected to the Internet has an IP address. To access a website, we need to enter its IP address on our web browser. But it is very difficult to remember the IP addresses of different websites as they are in terms of numbers or strings.

However, it is easier to remember names, and therefore, each computer server hosting a website or web resource is given a name against its IP address. These names are called the Domain names or hostnames corresponding to unique IP addresses assigned to each server. For easy understanding, it can be considered as the phonebook where instead of remembering each person's phone number, we assign names to their numbers. For example, IP addresses and domain names of some websites are as follows:

Table 10.1 Examples of domain names and their mapped IP addresses

Domain Name	IP Address
ncert.nic.in	164.100.60.233
cbse.nic.in	164.100.107.32
mhrd.gov.in	164.100.163.45
wikipedia.org	198.35.26.96

10.8.1 DNS Server

Instead of remembering IP addresses, we assign a domain name to each IP. But, to access a web resource, a browser needs to find out the IP address corresponding to the domain name entered. Conversion of the domain

© NCERT not to be republished

Reprint 2026-27

name of each web server to its corresponding IP address is called domain name resolution. It is done through a server called DNS server. Thus, when we enter a URL on a web browser, the HTTP protocol approaches a computer server called DNS server to obtain the IP address corresponding to that domain name. After getting the IP address, the HTTP protocol retrieves the information and loads it in our browser.

In Figure 10.20, an example is shown in which the HTTP requests a DNS server for corresponding IP addss, and the server sends back an IP address.

DNS root servers are named using alphabets A through M for the first 13 letters of the alphabet. Ten of these servers are in the US, one in London, one in Stockholm, and one in Japan. The organisation Internet Assigned Numbers Authority (IANA) keeps this list of DNS root servers.

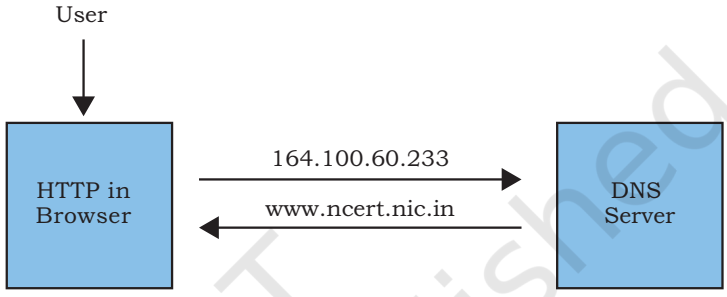


Figure 10.20: Request of IP address corresponding to domain name

A DNS server maintains a database of domain names and their corresponding IP addresses. To understand how the domain name resolution works, we have to understand how and where the DNS servers are kept. The DNS servers are placed in hierarchical order. At the top level, there are 13 servers called root servers. Then below the root servers there are other DNS servers at different levels. A DNS server may contain the IP address corresponding to a domain or it will contain the IP address of other DNS servers, where this domain entry can be searched.

Summary

- A computer network is an interconnection among two or more computers or computing devices.
- A computer network allows computers to share data and resources among each other.
- Networking devices are used to connect multiple computers in different settings.

© NCERT not to be republished

- In a communication network, each device that is a part of a network and that can receive, create, store or send data to different network routes is called a node.
- Based on the geographical area covered and data transfer rate, computer networks are broadly categorised into LAN (Local Area Network), MAN (Metropolitan Area Network) and WAN (Wide Area Network).
- LAN is a network that connects a variety of nodes placed at a limited distance ranging from a single room, a floor, an office or a campus having one or more buildings in the same premises.
- Ethernet is a set of rules that decides how computers and other devices connect with each other through cables in a LAN.
- Metropolitan Area Network (MAN) is an extended form of LAN which covers a larger geographical area like a city or a town.
- Cable TV network or cable based broadband internet services are examples of MAN.
- Wide Area Network (WAN) connects computers and other LANs and MANs, which are spread across different geographical locations of a country or in different countries or continents.
- The Internet is the largest WAN that connects billions of computers, smartphones and millions of LANs from different continents.
- Modem stands for 'MOdulator DEModulator', is a device used for conversion between electric signals and digital bits.
- Ethernet card, also known as Network Interface Card (NIC card in short) is a network adaptor used to set up a wired network.
- Each NIC has a MAC address, which helps in uniquely identifying the computer on the network.
- A repeater is an analog device that regenerate the signals on the cables to which it is connected.
- A switch is a networking device used to connect multiple computers or communicating devices.
- A router is a network device that can receive the data, analyse it and transmit it to other networks.

© NCERT not to be republished

Reprint 2026-27

- Gateway serves as the entry and exit point of a network, as all data coming in or going out of a network must first pass through the gateway in order to use routing paths.
- The arrangement of computers and other peripherals in a network is called its topology.
- Common network topologies are Mesh, Ring, Bus, Star and Tree.
- In mesh topology each communicating device is connected with every other device in the network.
- In ring topology, each node is connected to two other devices, one each on either side.
- In bus topology, a single backbone wire called bus is shared among the nodes, which makes it cheaper and easy to maintain.
- In star topology, each communicating device is connected to a central networking device like a hub or a switch.
- In tree or hybrid topology, there are multiple branches and each branch can have one or more basic topologies like star, ring and bus.
- The MAC address, also known as the physical or hardware address, is a unique permanent value associated with a network adapter called a NIC. It is used to physically identify a machine on the network.
- IP address, also known as Internet Protocol address, is a unique address that can be used to uniquely identify each node in a network.
- Unlike MAC address, IP address can change if a node is removed from one network and connected to another network.
- The Internet is the global network of computing devices.
- The World Wide Web (WWW) or web in short, is an ocean of information, stored in the form of trillions of interlinked web pages and web resources.
- Sir Tim Berners-Lee — a British computer scientist invented the revolutionary World Wide Web in 1990.
- HTML (HyperText Markup Language) is a language which is used to design standardised Web Pages so that the Web contents can be read

© NCERT not to be republished

and understood from any computer.

- URI (Uniform Resource Identifier) or URL (Uniform Resource Locator) is a unique address or path for each resource located on the web.
- HTTP – The HyperText Transfer Protocol is a set of rules which is used to retrieve linked web pages across the web. The more secure and advanced version is HTTPS.
- Each computer server hosting a website or web resource is given a name against its IP address. These names are called the Domain names or hostnames.
- Conversion of the domain name of each web server to its corresponding IP address is called domain name resolution. It is done through a server called DNS server.

EXERCISE

1. Expand the following:
   a) ARPANET
   b) MAC
   c) ISP
   d) URI
2. What do you understand by the term network?
3. Mention any two main advantages of using a network of computing devices.
4. Differentiate between LAN and WAN.
5. Write down the names of few commonly used networking devices.
6. Two universities in different States want to transfer information. Which type of network they need to use for this?
7. Define the term topology. What are the popular network topologies?
8. How is tree topology different from bus topology?
9. Identify the type of topology from the following:
   a) Each node is connected with the help of a single cable.
   b) Each node is connected with central switching through independent cables.

© NCERT not to be republished

Reprint 2026-27

10. What do you mean by a modem? Why is it used?
11. Explain the following devices:
    a) Switch
    b) Repeater
    c) Router
    d) Gateway
    e) NIC
12. Draw a network layout of star topology and bus topology connecting five computers.
13. What is the significance of MAC address?
14. How is IP address different from MAC address? Discuss briefly.
15. What is DNS? What is a DNS server?
16. Sahil, a class X student, has just started understanding the basics of Internet and web technologies. He is a bit confused in between the terms "World Wide Web" and "Internet". Help him in understanding both the terms with the help of suitable examples of each.

© NCERT not to be republished

Reprint 2026-27

Chapter 11

Data Communication

12130CH11

"People already have bionic arms and legs that work by the power of thought. And we increasingly outsource mental and communicative activities to computers. We are merging with our smartphones. Very soon, they will just be part of the body"

— Yuval Noah Harari

In this Chapter

- *Concept of Communication*
- *Components of Data Communication*
- *Measuring Capacity of Communication Media*
- *Types of Data Communication*
- *Switching Techniques*
- *Transmission Media*
- *Mobile Telecommunication Technologies*
- *Protocol*

11.1 Concept of Communication

The term "Data Communication" comprises two words: Data and Communication. Data can be any text, image, audio, video, and multimedia files. Communication is an act of sending or receiving data. Thus, data communication refers to the exchange of data between two or more networked or connected devices. These devices must be capable of sending and receiving data over a communication medium. Examples of such devices include personal computers, mobile phones, laptops, etc. As we can see in Figure 11.1, four different types of devices — computer, printer, server and switch are connected to form the network. These devices are connected through a media to the network, which carry information from one end to other end.

© NCERT not to be republished

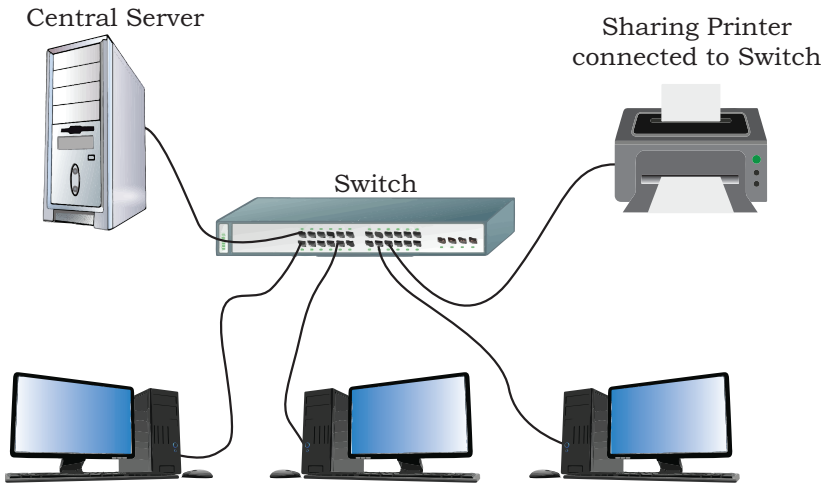


Figure 11.1: A simple network of computing devices

11.2 Components of Data Communication

Whenever we talk about communication between two computing devices using a network, five most important aspects come to our mind. These are sender, receiver, communication medium, the message to be communicated, and certain rules called protocols to be followed during communication. The communication media is also called transmission media. Figure 11.2 shows the role of these five components in data communication.

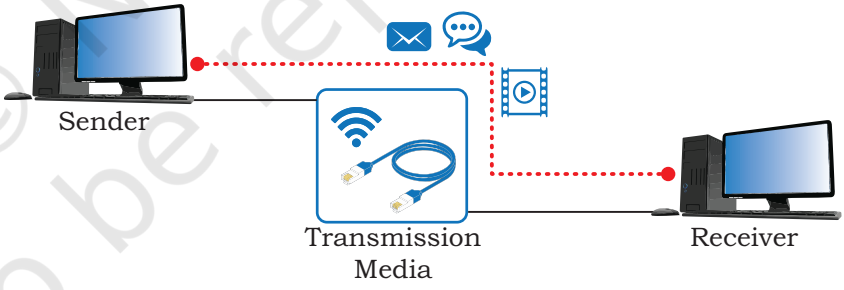


Figure 11.2: Components of data communication

Activity 11.1

List various types of senders on a network.

Sender: A sender is a computer or any such device which is capable of sending data over a network. It can be a computer, mobile phone, smartwatch, walkie-talkie, video recording device, etc.

Receiver: A receiver is a computer or any such device which is capable of receiving data from the network. It can be any computer, printer, laptop, mobile phone, television, etc. In computer communication, the sender and receiver are known as nodes in a network.

© NCERT not to be republished

Reprint 2026-27

Message: It is the data or information that needs to be exchanged between the sender and the receiver. Messages can be in the form of text, number, image, audio, video, multimedia, etc.

Communication media: It is the path through which the message travels between source and destination. It is also called medium or link which is either wired or wireless. For example, a television cable, telephone cable, ethernet cable, satellite link, microwaves, etc. We will study about various communication media in section 11.5.

Protocols: It is a set of rules that need to be followed by the communicating parties in order to have successful and reliable data communication. You have already come across protocols such as Ethernet and HTTP.

11.3 Measuring Capacity of Communication Media

In data communication, the transmission medium is also known as channel. The capacity of a channel is the maximum amount of signals or traffic that a channel can carry. It is measured in terms of bandwidth and data transfer rate as described below:

11.3.1 Bandwidth

Bandwidth of a channel is the range of frequencies available for transmission of data through that channel. Higher the bandwidth, higher the data transfer rate. Normally, bandwidth is the difference of maximum and minimum frequency contained in the composite signals. Bandwidth is measured in Hertz (Hz).

$$1\ KHz = 1000\ Hz$$
$$1\ MHz = 1000\ KHz = 1000000\ Hz$$

Activity 11.2

Find out how many hertz is 10 Megahertz.

11.3.2 Data Transfer Rate

Data travels in the form of signals over a channel. One signal carries one or more bits over the channel. Data transfer rate is the number of bits transmitted between source and destination in one second. It is also known as bit rate. It is measured in terms of bits per second (bps). The higher units for data transfer rates are:

$$1\ Kbps = 2^{10}\ bps = 1024\ bps$$
$$1\ Mbps = 2^{20}\ bps = 1024\ Kbps$$
$$1\ Gbps = 2^{30}\ bps = 1024\ Mbps$$
$$1\ Tbps = 2^{40}\ bps = 1024\ Gbps$$

MBps stands for Megabyte per second whereas Mbps stands for Megabit per second.

© NCERT not to be republished

Example 11.1 A user wants to upload a text document at the rate of 10 pages per 20 second. What will be the required data rate of the channel? (Assume that 1 page contains 1600 characters and each character is of 8 bits).

Solution: Required data rate $\frac{(10 \times 1600 \times 8)}{20}$

$= 6400\,bps = 6.25\,Kbps$

11.4 Types of Data Communication

Data communication happens in the form of signals between two or more computing devices or nodes. The transfer of data happens over a point-to-point or multipoint communication channel. Data communication between different devices are broadly categorised into 3 types: Simplex communication, Half-duplex communication, and Full-duplex communication.

11.4.1 Simplex Communication

It is a one way or unidirectional communication between two devices in which one device is sender and other one is receiver. Devices use the entire capacity of the link to transmit the data. It is like a one way street where vehicles can move in only one direction. For example, data entered through a keyboard or audio sent to a speaker are one way communications.

With the advent of IoT, controlling home appliances is another example of simplex communication as shown in the Figure 11.3. One can control fans, lights, fridge, oven etc. while sitting in the office or driving a car.

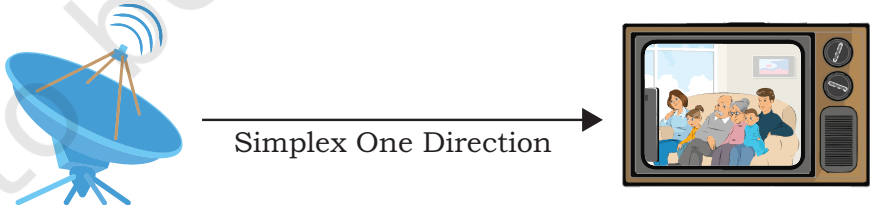


Figure 11.3: Simplex communication

11.4.2 Half-duplex Communication

It is two way or bidirectional communication between two devices in which both the devices can send and receive data or control signals in both directions, but not at the same time, as shown in Figure 11.4. While one device is sending data, the other one will receive and vice-versa. It is like sharing a one-way narrow bridge among vehicles

© NCERT not to be republished

moving in both directions. Vehicles cannot pass the bridge simultaneously. Basically, it is a simplex channel where the direction of transmission can be switched. Application of such type of communication can be found in walkie-talkie where one can press the push-to-talk button and talk. This enables the transmitter and turns off the receiver in that device and others can only listen.

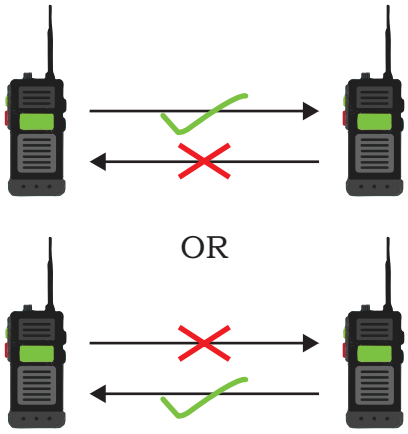


Figure 11.4: Half-duplex where communication occurs in two different moments.

11.4.3 Full-duplex Communication

It is two way or bidirectional communication in which both devices can send and receive data simultaneously, as shown in Figure 11.5. It is like a two way road where vehicles can go in both directions at the same time. This type of communication channel is employed to allow simultaneous communication, for example, in our mobile phones and landline telephones. The capacity of the transmission link is shared between the signals going in both directions. This can be done either by using two physically separate simplex lines — one for sending and other for receiving, or the capacity of the single channel is shared between the signals travelling in different directions.

Figure 11.5: Full duplex transmission of data

© NCERT not to be republished

Reprint 2026-27

11.5 Switching Techniques

In a network having multiple devices, we are interested to know how to connect the sender and receiver so that one-to-one communication is possible. One solution is to make a dedicated connection between each pair of devices (mesh topology) or between a central device and every other device (a star topology). However, we know that such methods are costly in case of large networks.

An alternative to this is switching whereby data is routed through various nodes in a network. This switching process forms a temporary route for the data to be transmitted. Two commonly used switching techniques are — Circuit Switching and Packet Switching.

VoIP is a communication methodology designed to deliver both voice and multimedia communications over Internet protocol.

11.5.1 Circuit Switching

In circuit switching, before a communication starts, a dedicated path is identified between the sender and the receiver. This path is a connected sequence of links between network nodes. All packets follow the same path established during the connection.

In earlier days, when we placed a telephone call, the switching equipment within the telephone system finds out a physical path or channel all the way from our telephone at home to the receiver's telephone. This is an example of circuit switching.

Voice over Long-Term Evolution (VoLTE) is a standard for high-speed wireless communication for mobile phones, including IoT and wearables.

11.5.2 Packet Switching

In packet switching, each information or message to be transmitted between sender and receiver is broken down into smaller pieces, called packets. These packets are then transmitted independently through the network. Different packets of the same message may take different routes depending on availability.

Each packet has two parts — a header containing the address of the destination and other information, and the main message part. When all the packets reach the destination, they are reassembled and the complete message is received by the receiver.

Unlike circuit switching, a channel is occupied in packet switching only during the transmission of the packet. On completion of the transmission, the channel is available for transfer of packets from other communicating parties.

© NCERT not to be republished

11.6 TRANSMISSION MEDIA

A transmission medium can be anything that can carry signals or data between the source (transmitter) and destination (receiver). For example, as we switch on a ceiling fan or a light bulb, the electric wire is the medium that carries electric current from switch to the fan or bulb. Two men are talking as shown in Figure 11.6. Here the medium is air.

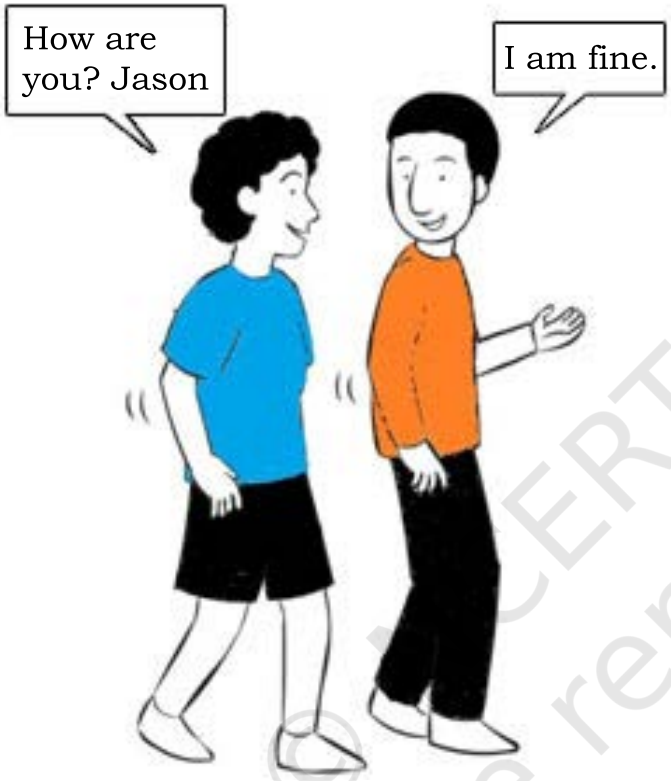


Figure 11.6: Two person communicating

In data communication, transmission media are the links that carry messages between two or more communicating devices. Transmission can be classified as guided or unguided. Figure 11.7 shows the classification of communication media.

In guided transmission, there is a physical link made of wire/cable through which data in terms of signals are propagated between the nodes. These are usually metallic cable, fiber-optic cable, etc. They are also known as wired media.

In unguided transmission, data travels in air in terms of electromagnetic waves using an antenna. They are also known as wireless media.

© NCERT not to be republished

Reprint 2026-27

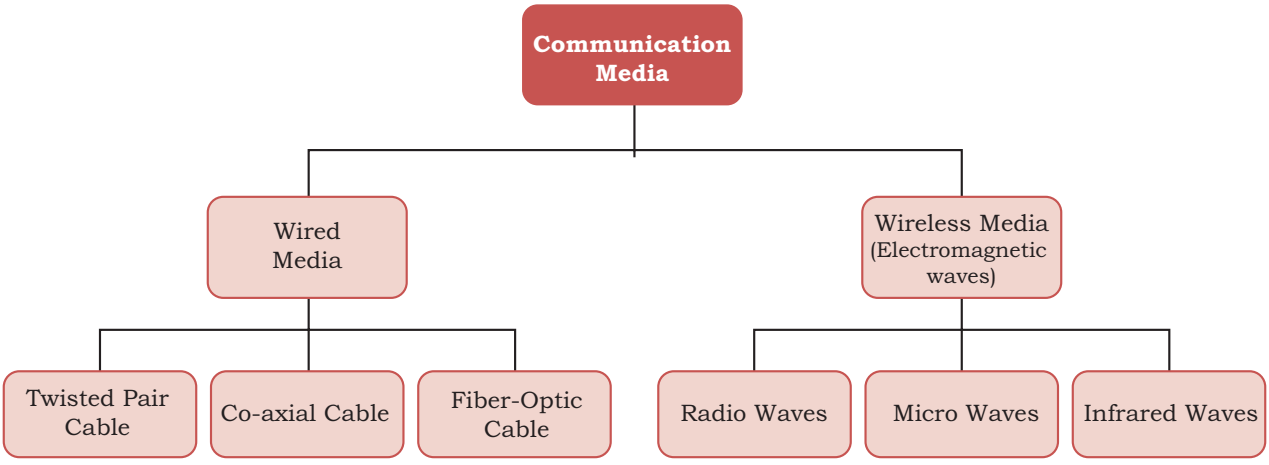


Figure 11.7: Classification of communication media

Dish-shaped antennas are used for sending and receiving data at longer distances. These antennas are mounted on taller buildings so that it would be in line-of-sight. Waves gradually become weaker and weaker after travelling a certain distance through the air. Therefore repeaters are installed to regenerate the signals of the same energy.

11.6.1 Wired Transmission Media

Any physical link that can carry data in the form of signals belongs to the category of wired transmission media. Three commonly used guided/wired media for data transmission are, twisted pair, coaxial cable, and fiber optic cable. Twisted-pair and coaxial cable carry the electric signals whereas the optical fiber cable carries the light signals.

(A) Twisted Pair Cable

A twisted-pair consists of two copper wires twisted like a DNA helical structure. Both the copper wires are insulated with plastic covers. Usually, a number of such pairs are combined together and covered with a protective outer wrapping, as shown in Figure 11.8.

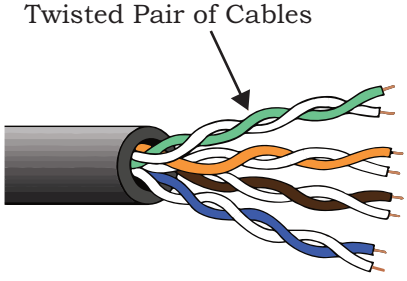


Figure 11.8: Twisted pair of cables

© NCERT not to be republished

Each of the twisted pairs act as a single communication link. The use of twisted configuration minimises the effect of electrical interference from similar pairs close by. Twisted pairs are less expensive and most commonly used in telephone lines and LANs. These cables are of two types: Unshielded twisted-pair (UTP) and Shielded twisted-pair (STP), as shown in Figure 11.9.

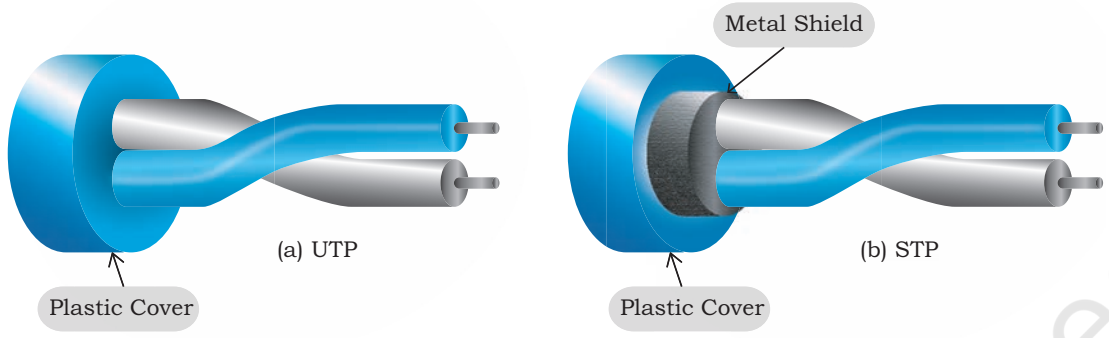


Figure 11.9: UTP Cable and STP Cable

(B) Coaxial cable

Coaxial cable is another type of data transmission medium. It is better shielded and has more bandwidth than a twisted pair. As shown in Figure 11.10, it has a copper wire at the core of the cable which is surrounded with insulating material. The insulator is further surrounded with an outer conductor (usually a copper mesh). This outer conductor is wrapped in a plastic cover. The key to success of coaxial cable is its shielded design that allows the cable's copper core to transmit data quickly, without interference of environmental factors. These types of cables are used to carry signals of higher frequencies to a longer distance.

Backbone networks interconnect different segments of the network and provide a path to exchange information between different LANs or subnetworks..

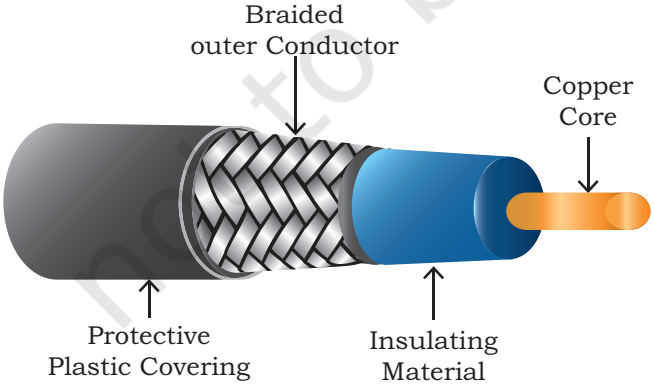


Figure 11.10: A coaxial cable

© NCERT not to be republished

The number of oscillations a wave makes per second is called its frequency, and it is measured in Hz (Hertz).

(C) Optical Fibre

The optical fiber cable carries data as light, which travels inside a thin fiber of glass (Figure 11.11). Optic fiber uses refraction to direct the light through the media. A thin transparent strand of glass at the centre is covered with a layer of less dense glass called cladding. This whole arrangement is covered with an outer jacket made of PVC or Teflon. Such types of cables are usually used in backbone networks. These cables are of light weight and have higher bandwidth which means higher data transfer rate. Signals can travel longer distances and electromagnetic noise cannot affect the cable. However, optic fibers are expensive and unidirectional. Two cables are required for full duplex communication.

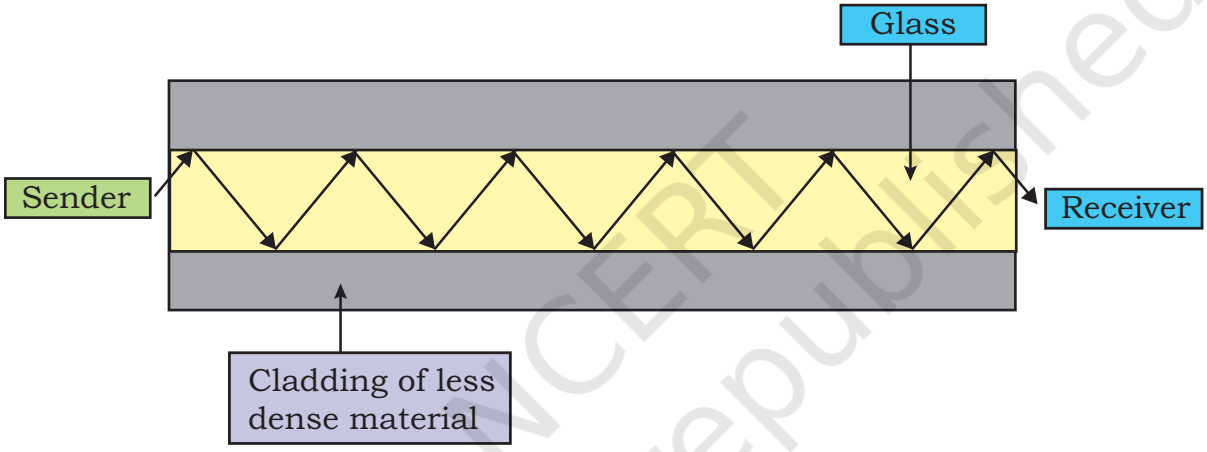


Figure 11.11: Fiber optic cable

Geostationary satellites orbiting around the Earth are used to deliver broadband Internet service, similar to the way satellite is used for television and telephone signals. These satellites use microwaves for communication between a satellite dish placed at our home and the hub of satellite internet service providers.

11.6.2 Wireless Transmission Media

In wireless communication technology, information In wireless communication technology, information travels in the form of electromagnetic signals through air. Electromagnetic spectrum of frequency ranging from 3 KHz to 900 THz is available for wireless communication (Figure 11.12). Wireless technologies allow communication between two or more devices in short to long distance without requiring any physical media. There are many types of wireless communication technologies such as Bluetooth, WiFi, WiMax etc.

The electromagnetic spectrum range (3KHz to 900THz) can be divided into 4 categories (Figure 11.12) - Radio waves, Microwaves, Infrared waves and Visible or Light waves, according to their frequency ranges. Some

© NCERT not to be republished

Reprint 2026-27

of the properties of each wave are listed in Table 11.1. of these, three are useful for wireless communication.

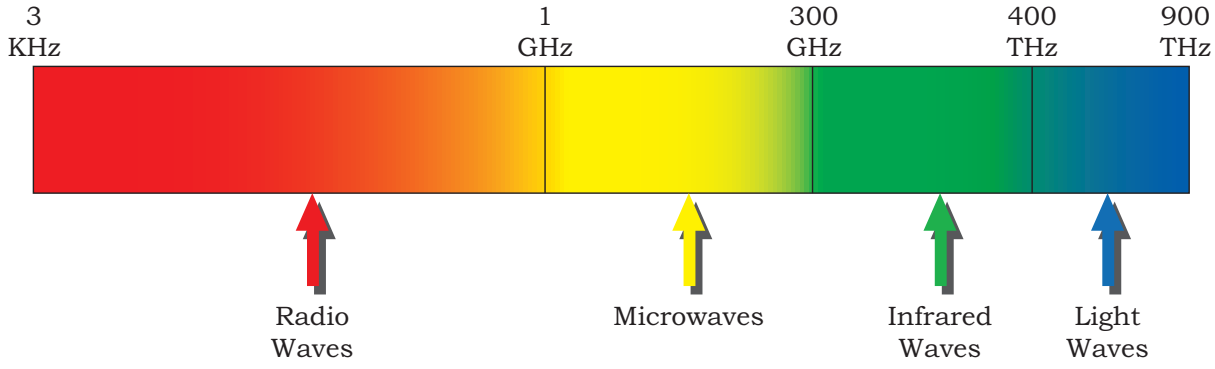


Figure 11.12: Electromagnetic waves spectrum

Table 11.1 Classification of transmission waves and their properties

Transmission Waves	Properties
Radio Waves	1. Waves of frequency range 3 KHz - 1 GHz 2. Omni-directional, these waves can move in all directions 3. Radio waves of frequency 300KHz-30MHz can travel long distance 4. Susceptible to interference 5. Radio waves of frequency 3-300KHz can penetrate walls 6. These waves are used in AM and FM radio, television, cordless phones.
Microwaves	1. Electromagnetic waves of frequency range 1GHz - 300GHz. 2. Unidirectional, can move in only one direction. 3. Cannot penetrate solid objects such as walls, hills or mountains. 4. Needs line-of-sight propagation i.e. both communicating antenna must be in the direction of each other. 5. Used in point-to-point communication or unicast communication such as radar and satellite. 6. Provide very large information-carrying capacity.
Infrared waves	1. Electromagnetic waves of frequency range 300GHz - 400THz. 2. Very high frequency waves. 3. Cannot penetrate solid objects such as walls. 4. Used for short-distance point-to-point communication such as mobile-to-mobile, mobile-to-printer, remote-control-to-TV, and Bluetooth-enabled devices to other devices like mouse, keyboards etc.

11.6.3 Wireless Technologies

(A) Bluetooth

Bluetooth is a short-range wireless technology that can be used to connect mobile-phones, mouse, headphones, keyboards, computers, etc. wirelessly over a short distance. One can print documents with bluetooth-

© NCERT not to be republished

enabled printers without a physical connection. All these bluetooth-enabled devices have a low cost transceiver chip. This chip uses the unlicensed frequency band of 2.4 GHz to transmit and receive data. These devices can send data within a range of 10 meters with a speed of 1 - 2 Mbps.

In Bluetooth technology, the communicating devices within a range of 10 meters build a personal area network called piconet. The devices in a piconet work in a master-slave configuration. A master device can communicate with up to 7 active slave devices at the same time.

Bluetooth technology allows up to 255 devices to build a network. Out of them, 8 devices can communicate at the same time and remaining devices can be inactive, waiting for a response command from the master device.

(B) Wireless LAN

This is another way of wireless communication. Wireless LAN is a local area network (LAN), and it is a popular way to connect to the Internet. The international organisation IEEE assigns numbers to each different standards of LAN. The wireless LAN is number as 802.11, and it is popularly known as Wi-Fi.

These networks consist of communicating devices such as laptops and mobile phones, as well as the network device called APs (access points) which is installed in buildings or floors (Figure 11.13). An access point is a device that is used to create a wireless local area network, by connecting to a wired router, switch, or hub. The APs are connected to a wired network, and all the devices communicate or access the Internet through an access point.

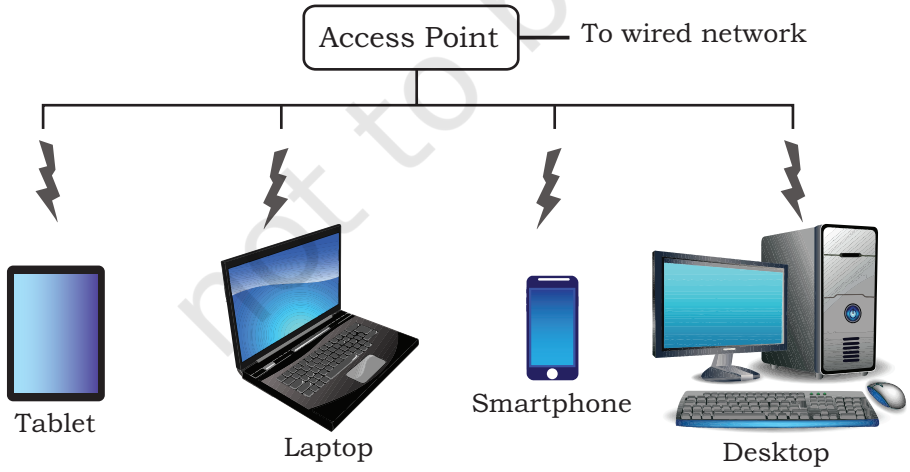


Figure 11.13: Access point creating a wireless LAN

Wi-Fi gives users the flexibility to move around within the network area while being connected to the network. Following are some of the benefits of WLAN:

© NCERT not to be republished

- Wireless connections can be used to extend or replace an existing wired infrastructure
- Resulted in increased access for mobile devices
- Provides easy access to the Internet in public places

11.7 Mobile Telecommunication Technologies

Today the mobile phone network is the most used network in the world. The ability to be connected to the network on-the-go makes it very convenient to communicate with people via call or instant messages. It is also handy to access the Internet using the mobile phone network through wireless connection. Besides, the Internet of Things (IoT) is letting us control and communicate with other smart devices as well.

The architecture of the mobile network has rapidly evolved over the last few decades. The different landmark achievements in mobile communication technologies are classified as different generations. They are identified as 1G, 2G, 3G, 4G, and 5G. Let us briefly discuss the mobile telecommunication generations.

The first generation (1G) mobile network system came around 1982. It was used to transmit only voice calls. The analog signals were used to carry voices between the caller and receiver.

The second generation (2G) mobile network system came around 1991. Instead of analog signals, voice calls were transmitted in digital form thus providing improved call quality. This increased capacity allowed more people to talk simultaneously, and led to improved security as the signals could be encrypted. It also enabled an additional service to send SMS and MMS (Multimedia messages).

The third generation (3G) mobile network technology was developed during late 90s, but it was introduced commercially around 2001. It offered both digital voice and data services. 3G provided Internet access via the same radio towers that provide voice service to the mobile phone. It facilitated greater voice and data capacity. Therefore, more simultaneous calls could happen in the same frequency range and also a significantly faster data transfer speed.

Demand for faster data is always increasing and thus 4G mobile networks were developed and now

WiMax stands for Worldwide Interoperability for Microwave Access. Like Wi-Fi, it is also used for communication in wireless networks but there is a difference. Whereas Wi-Fi is used to form small wireless networks (WLANs), WiMax uses a larger spectrum to deliver connections to various devices on the network. It has a higher data transfer rate and can span over a larger area. That is why it is used in MAN applications.

© NCERT not to be republished

Reprint 2026-27

Think and Reflect

Explore how 5G can impact society.

5G networks have also come into being. 4G is much faster than 3G and this has revolutionised the field of telecommunication by bringing the wireless experience to a new level altogether. 4G systems support interactive multimedia, voice, video, wireless internet and other broadband services. Technologically, 4G is very different compared to 3G.

The fifth generation or 5G is currently under development. It is expected to be a milestone development for the success of IoT and Machine to Machine (M2M) communications. Machine to machine (M2M) is direct communication between devices — wired and wireless. 5G is expected to allow data transfer in Gbps, which is much faster than 4G. It is expected to be able to support all the devices of the future such as connected vehicles and the Internet of Things.

11.8 Protocol

In communication, Protocol is a set of standard rules that the communicating parties — the sender, the receiver, and all other intermediate devices need to follow. We know that the sender and receiver can be parts of different networks, placed at different geographic locations. Besides, the data transfer rates in different networks can vary, requiring data to be sent in different formats.

11.8.1 Need for Protocols

We need protocols for different reasons such as flow control, access control, addressing, etc. Flow control is required when the sender and receiver have different speeds of sending and receiving the data. Figure 11.14 shows that Computer A is sending data at the speed of 1024 Mbps and computer B is receiving data at the speed of 512 Mbps. In this case, Computer B must be able to inform computer A about the speed mismatch so that computer A can adjust its data transmission rate. Otherwise some data will be lost, as shown in Figure 11.14.

Access control is required to decide which nodes in a communication channel will access the link shared among them at a particular instant of time. Otherwise, the transmitted data packets will collide if computers are sending data simultaneously through the same link resulting in the loss or corruption of data.

© NCERT not to be republished

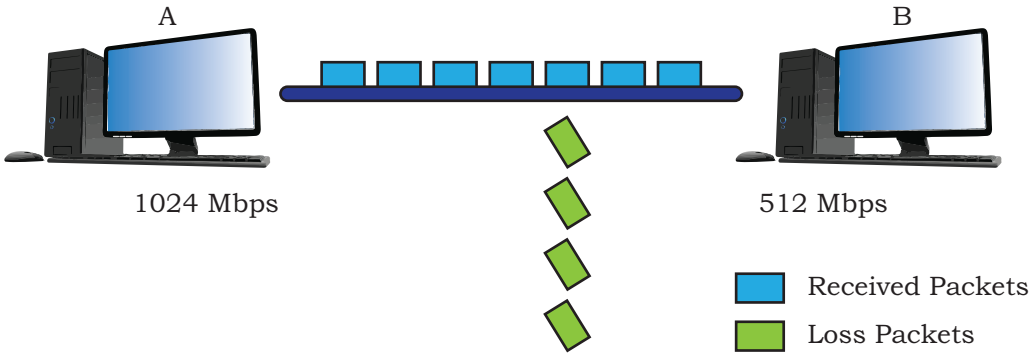


Figure 11.14: Speed mismatch between two computers can result into loss of data

Protocols also define:

- how computers identify one another on a network.
- the form to which the data should be converted for transit.
- how to decide whether the data received is for that node or to be forwarded to another node.
- ensuring that all the data have reached the destination without any loss.
- how to rearrange the packets and process them at the destination.

If all the rules or protocols of a communication network are defined at one place, it becomes complex to ensure that communicating parties follow the guidelines. In this section, we will briefly talk about some of the protocols required in communication.

11.8.2 HyperText Transfer Protocol (HTTP)

HTTP stands for HyperText Transfer Protocol. It is the primary protocol used to access the World Wide Web. Tim Berners-Lee led the development of HTTP at CERN in 1989 in collaboration with Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).

Hypertext refers to a document that contains images or text that can be linked to another document or text.

HTTP is a request-response (also called client-server) protocol that runs over TCP. The common use of HTTP is between a web browser (client) and a web server (server). HTTP facilitates access of hypertext from the World Wide Web by defining how information are formatted and transmitted, and how the Web servers and browsers should respond to various commands.

© NCERT not to be republished

A web page is written using a markup language like HTML and is stored on a web server for access via its URL. Once a user opens a web browser and types in the URL of the intended web page, a logical communication link between the user machine (client) and the web server is created using HTTP.

For example, whenever we enter the URL http//www.ncert.nic.in in a browser, it sends HTTP request to the web-server where ncert.nic.in is hosted. The HTTP response from the web-server fetches and sends the requested Web-page, which is displayed on your browser.

11.8.3 File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is the protocol used for transferring files from one machine to another. Like HTTP, FTP also works on a client-server model.

When a user requests for a file transfer with another system, FTP sets up a connection between the two nodes for accessing the file. Optionally, the user can authenticate using user ID and password. The user then specifies the file name and location of the desired file. After that, another connection sets up and the file transfer happens directly between the two machines. However, some servers provide FTP logins without authentication for accessing files.

File transfer between two systems seems simple and straightforward because FTP takes care of issues between two communicating devices, such as:

- use of different conventions while naming files.
- representation of text and data in different formats.
- having different directory structure

11.8.4 Point to Point Protocol (PPP)

PPP is a communication protocol which establishes a dedicated and direct connection between two communicating devices. This protocol defines how two devices will authenticate each other and establish a direct link between them to exchange data. For example, two routers with direct connection communicate using PPP. The Internet users who connect their home computers to the server of an Internet Service Provider (ISP) through a modem also use PPP.

The communicating devices should have duplex modes for using this protocol. This protocol maintains

© NCERT not to be republished

data integrity ensuring that the packets arrive in order. It intimates the sender about damage or lost packets and asks to resend it.

11.8.5 Simple Mail Transfer Protocol (SMTP)

SMTP is a protocol used for email services. It uses information written on the message header (like an envelope on a letter sent by post), and is not concerned with the content of the email message. Each email header contains email addresses of recipients. The email containing header and body are entered into a queue of outgoing mails.

The SMTP sender program takes mails from the outgoing queue and transmits them to the destination(s). When the SMTP sender successfully delivers a particular mail to one or more destinations, it removes the corresponding receiver's email address from the mail's destination list. When that mail is delivered to all the recipients, it is removed from the outgoing queue. The SMTP receiver program accepts each mail that has arrived and places it in the appropriate user mailbox.

Activity 11.3

Find and list other Email Protocols.

11.8.6 Transmission Control Protocol (TCP)/ Internet Protocol (IP)

TCP/IP stands for Transmission Control Protocol/ Internet Protocol. It is a set of standardised rules that uses a client-server model of communication in which a user or machine (a client) requests a service by a server in the network.

The IP protocol ensures that each computer or node connected to the Internet is assigned an IP address, which is used to identify each node independently. It can be considered to be the adhesive that holds the whole Internet together. TCP ensures that the message or data is broken into smaller chunks, called IP packets. Each of these packets are routed (transmitted) through the Internet, along a path from one router to the next, until it reaches the specified destination. TCP guarantees the delivery of packets on the designated IP address. It is also responsible for ordering the packets so that they are delivered in sequence.

There are many redundant connection paths in the Internet, with backbones and ISPs connecting to each other in multiple locations. So, there are many

© NCERT not to be republished

Reprint 2026-27

possible paths between two hosts. Hence, two packets of the same message can take two different routes depending on congestion and other factors in different possible routes. When all the packets finally reach the destination machine, they are reassembled into the original message at the receiver's end.

SUMMARY

- Data communication refers to the exchange of data between two or more networked or connected devices like laptops, PC, printers, routers etc.
- Sender, receiver, messages, channel and protocols are major components of data communication.
- In data communication, transmission media are the links that carry messages between two or more communicating devices. These are broadly classified into guided and unguided media.
- In guided transmission, there is a physical link made of wire/cable through which data in terms of signals are propagated between the nodes. These are usually metallic cable, fiber-optic cable, etc. They are also known as wired media.
- In unguided transmission, data travels in air in terms of electromagnetic waves using an antenna. They are also known as wireless media.
- The capacity of channels is measured in bandwidth. The unit of bandwidth is Hertz.
- Communication can be done in three different modes — simplex, half-duplex, and full-duplex communication.
- Switching techniques are alternative to dedicated lines whereby data is routed through various nodes in a network. It forms a temporary route for the data to be transmitted. Two commonly used switching techniques are – circuit switching and packet switching.
- Electromagnetic spectrum of frequency ranging from 3 KHz to 900 THz is available for wireless communication. This spectrum range (3KHz to 900THz) can be divided into four categories- Radio

© NCERT not to be republished

waves, Microwaves, Infrared waves and Visible or Light waves, according to their frequency ranges.

- Bluetooth is a short-range wireless technology that can be used to connect mobile-phones, mouse, headphones, keyboards, computers, etc. wirelessly over a short distance.
- Based on the architecture of the mobile network, mobile communication technologies are classified into different generations identified as 1G, 2G, 3G, 4G, and 5G.
- In communication, protocol is a set of standard rules that the communicating parties — the sender, the receiver, and all other intermediate devices need to follow. Flow control, access control, addressing, etc. are examples of protocol.
- HTTP stands for HyperText Transfer Protocol. It is the primary protocol used to access the World Wide Web, which was developed by Tim Berners-Lee at CERN in 1989.
- File Transfer Protocol (FTP) is the protocol used for transferring files from one machine to another. Like HTTP, FTP also works on a client-server model.
- Point-to-Point protocol (PPP) defines how two devices will authenticate each other and establish a direct link between them to exchange data.
- TCP/IP stands for Transmission Control Protocol/ Internet Protocol. It is a set of standardised rules that uses a client-server model of communication in which a user or machine (a client) requests a service by a server in the network.

EXERCISE

1. What is data communication? What are the main components of data communication?
2. Which communication mode allows communication in both directions simultaneously?
3. Among LAN, MAN, and WAN, which has the highest speed and which one can cover the largest area?

© NCERT not to be republished

Reprint 2026-27

4. What are three categories of wired media? Explain them.
5. Compare wired and wireless media.
6. Which transmission media carries signals in the form of light?
7. List out the advantages and disadvantages of optical fiber cable.
8. What is the range of frequency for radio waves?
9. 18 Gbps is equal to how many Bits per second?
10. HTTP stands for?
11. Write short note on the following:
    a) HTTP
    b) Bandwidth
    c) Bluetooth
    d) DNS
    e) Data transfer rate
12. What is protocol in data communication? Explain with an example.
13. A composite signal contains frequencies between 500 MHz and 1GHz. What is the bandwidth of a signal?

© NCERT not to be republished

Chapter

12 Security Aspects

12130CH12

"Treat your password like your toothbrush. Don't let anybody else use it, and get a new one every six months."

— *Clifford Stoll*

In this Chapter

- *Threats and Prevention*
- *Malware*
- *Antivirus*
- *Spam*
- *HTTP vs HTTPS*
- *Firewall*
- *Cookies*
- *Hackers and Crackers*
- *Network Security Threats*

12.1 Threats and Prevention

Being alone is the most ideal situation for an individual in terms of security. It applies to computers as well. A computer with no link to an external device or computer is free from the security threats arising otherwise. However, it is not an ideal solution for a human being or a computer to stay aloof in order to mitigate any security threats, as the world at present is on its way to become fully connected. This connectedness of various devices and computers has brought into our focus the various network threats and its prevention.

Network security is concerned with protection of our device as well as data from illegitimate access or misuse. Threats include all the ways in which one can exploit any vulnerability or weakness in a network or communication system in order to cause harm or damage one's reputation.

© NCERT not to be republished

12.2 MALWARE

Malware is a short term used for MALicious softWARE. It is any software developed with an intention to damage hardware devices, steal data, or cause any other trouble to the user. Various types of malware have been created from time-to-time, and large-scale damages have been inflicted. Many of these malware programs have been identified and counter measures have been initiated. However, different types of malware keep on coming on a regular basis that compromise the security of computer systems and cause intangible damages. Besides, each year, malware incur financial damages worth billions of dollars worldwide. Viruses, Worms, Ransomware, Trojans, and Spyware are some of the kinds of malware.

12.2.1 Virus

The term computer virus was coined by Fred Cohen in 1985 and has been borrowed from biological science with almost similar meaning and behavior, the only difference is that the victim is a computer system and the virus is a malicious software. A virus is a piece of software code created to perform malicious activities and hamper resources of a computer system like CPU time, memory, personal files, or sensitive information.

Mimicking the behaviour of a biological virus, the computer virus spreads on contact with another system, i.e. a computer virus infects other computer systems that it comes into contact with by copying or inserting its code into the computer programs or software (executable files). A virus remains dormant on a system and is activated as soon as the infected file is opened (executed) by a user.

Viruses behave differently, depending upon the reason or motivation behind their creation. Some of the most common intentions or motives behind viruses include stealing passwords or data, corrupting files, spamming the user's email contacts, and even taking control of the user's machine. Some well-known viruses include CryptoLocker, ILOVEYOU, MyDoom, Sasser and Netsky, Slammer, Stuxnet, etc.

12.2.2 Worms

The Worm is also a malware that incurs unexpected or damaging behaviour on an infected computer system. The major difference between a worm and a virus is that

© NCERT not to be republished

unlike a virus, a worm does not need a host program or software to insert its code into. Worms are standalone programs that are capable of working on its own. Also, a virus needs human triggering for replication (i.e. when a user opens/executes the infected file), while a worm replicates on its own and can spread to other computers through the network. Some prominent examples of worms include Storm Worm, Sobig, MSBlast, Code Red, Nimda, Morris Worm, etc.

12.2.3 Ransomware

It is a type of malware that targets user data. It either blocks the user from accessing their own data or threatens to publish the personal data online and demands ransom payment against the same. Some ransomware simply block the access to the data while others encrypt data making it very difficult to access. In May 2017, a ransomware WannaCry infected almost 200,000 computers across 150 countries. It worked by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency. It literally made its victims "cry" and hence the name.

Figure 12.1: A ransomware

12.2.4 Trojan

Since the ancient Greeks could not infiltrate the city of Troy using traditional warfare methods, they gifted the king of Troy with a big wooden horse with hidden soldiers inside and eventually defeated them. Borrowing

© NCERT not to be republished

the concept, a Trojan is a malware, that looks like a legitimate software and once it tricks a user into installing it, it acts pretty much like a virus or worm. However, a Trojan does not self-replicate or infect other files, it spreads through user interaction such as opening an email attachment or downloading and executing a file from the Internet. Some Trojans create backdoors to give malicious users access to the system.

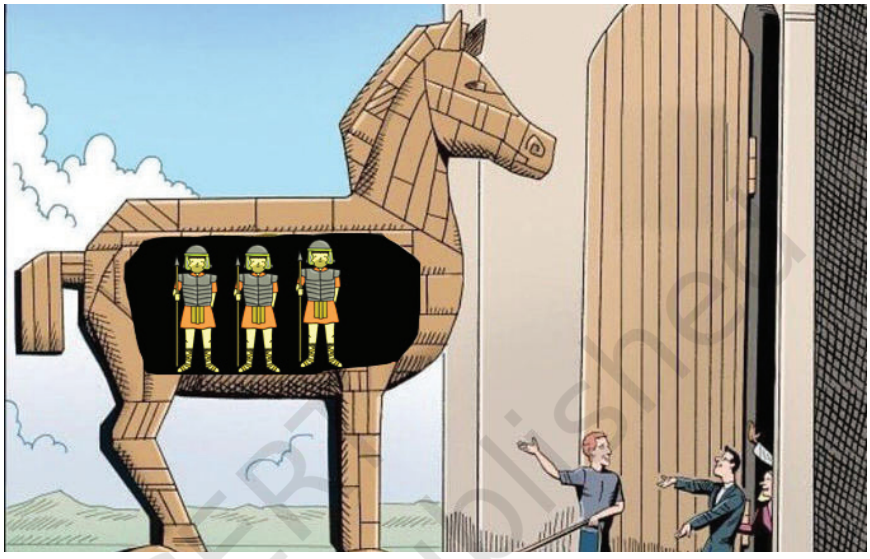


Figure 12.2: A trojan horse

12.2.5 Spyware

It is a type of malware that spies on a person or an organisation by gathering information about them, without the knowledge of the user. It records and sends the collected information to an external entity without consent or knowledge of the user.

Spyware usually tracks internet usage data and sells them to advertisers. They can also be used to track and capture credit card or bank account information, login and password information or user's personal identity.

12.2.6 Adware

An Adware is a malware that is created to generate revenue for its developer. An adware displays online advertisements using pop-ups, web pages, or installation screens. Once an adware has infected a substantial number of computer systems, it generates revenue either by displaying advertisements or using "pay per click" mechanism to charge its clients against the number of clicks on their displayed ads. Adware

© NCERT not to be republished

is usually annoying, but harmless. However, it often paves way for other malware by displaying unsafe links as advertisements.

To implement a keylogger in hardware, a thin transparent keyboard is placed atop the actual keyboard or input pad of the intended machine, which then records the keystrokes pressed by the user.

12.2.7 Keyloggers

A keylogger can either be malware or hardware. The main purpose of this malware is to record the keys pressed by a user on the keyboard. A keylogger makes logs of daily keyboard usage and may send it to an external entity as well. In this way, very sensitive and personal information like passwords, emails, private conversations, etc. can be revealed to an external entity without the knowledge of the user. One strategy to avoid the threat of password leaks by keyloggers is to use a virtual keyboard while signing into your online accounts from an unknown computer.

(A) Online Virtual Keyboard Vs On-Screen Keyboard

The names "on-screen" and "virtual" keyboard refer to any software-based keyboard and are sometimes used interchangeably. But, there exists a notable difference between "on-screen" and "online virtual" keyboards. Both types of keyboards may look the same, but the difference is in terms of the layout or ordering of the keys. The on-screen keyboard of an operating system uses a fixed QWERTY key layout (Figure 12.3), which can be exploited by sophisticated keylogger software. However, an online virtual keyboard randomises the key layout every time it is used (Figure 12.4), thereby making it very difficult for a keylogger software to know or record the key(s) pressed by the user.

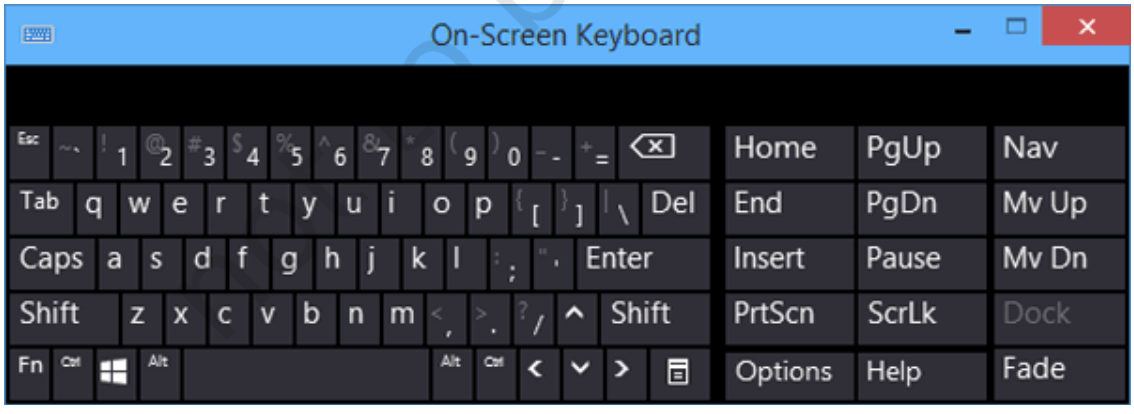


Figure 12.3: A QWERTY *keyboard layout*

© NCERT not to be republished

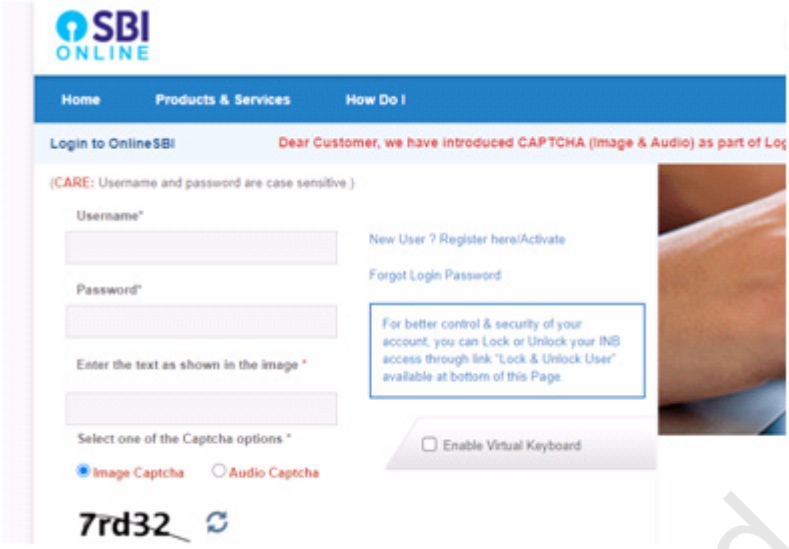


Figure 12.4: Online virtual keyboard

12.2.8 Modes of Malware distribution

A malware once designed, can take many routes to reach your computer. Some of the common distribution channels for malware are:

- **Downloaded from the Internet:** Most of the time, malware is unintentionally downloaded into the hard drive of a computer by the user. Of course, the malware designers are smart enough to disguise their malware, but we should be very careful while downloading files from the Internet (especially those highlighted as free stuff).
- **Spam Email:** We often receive an unsolicited email with embedded hyperlinks or attachment files. These links or attached files can be malware.
- **Removable Storage Devices:** Often, the replicating malware targets the removable storage media like pen drives, SSD cards, music players, mobile phones, etc. and infect them with malware that gets transferred to other systems that they are plugged into.
- **Network Propagation:** Some malware like Worms have the ability to propagate from one computer to another through a network connection.

12.2.9 Combating Malware

Common signs of some malware infection include the following:

© NCERT not to be republished

- frequent pop-up windows prompting you to visit some website and/or download some software;
- changes to the default homepage of your web browser;
- mass emails being sent from your email account;
- unusually slow computer with frequent crashes;
- unknown programs startup as you turn on your computer;
- programs opening and closing automatically;
- sudden lack of storage space, random messages, sounds, or music start to appear;
- programs or files appear or disappear without your knowledge.

Malware exists and continues to evolve, and so is the mechanism to combat them. As the saying goes that prevention is better than cure, we list some preventive measures against the malware discussed earlier.

- ✓ Using antivirus, anti-malware, and other related software and updating them on a regular basis.
- ✓ Configure your browser security settings
- ✓ Always check for a lock button in the address bar while making payments.
- ✓ Never use pirated on unlicensed software. Instead go for Free and Open Source Software (FOSS).
- ✓ Applying software updates and patches released by its manufacturers.
- ✓ Taking a regular backup of important data.
- ✓ Enforcing firewall protection in the network.
- ✓ Avoid entering sensitive (passwords, pins) or personal information on unknown or public computers.
- ✓ Avoid entering sensitive data on an unknown network (like Wi-Fi in a public place) using your own computer also.
- ✓ Avoid clicking on links or downloading attachments from unsolicited emails.
- ✓ Scan any removable storage device with an antivirus software before transferring data to and from it.
- ✓ Never share your online account or banking password/pins with anyone.
- ✓ Remove all the programs that you don't recognise from your system.

© NCERT not to be republished

Reprint 2026-27

- ✓ Do not install an anti-spyware or antivirus program presented to you in a pop-up or ad.
- ✓ Use the pop-up window's 'X' icon located on the top-right of the popup to close the ad instead of clicking on the 'close' button in the pop-up. If you notice an installation has been started, cancel immediately to avoid further damage.

12.3 Antivirus

Antivirus is a software, also known as anti-malware. Initially, antivirus software was developed to detect and remove viruses only and hence the name anti-virus. However, with time it has evolved and now comes bundled with the prevention, detection, and removal of a wide range of malware.

12.3.1 Methods of Malware Identification used by Antivirus

(A) Signature-based detection

In this method, an antivirus works with the help of a signature database known as "Virus Definition File (VDF)". This file consists of virus signatures and is updated continuously on a real-time basis. This makes the regular update of the antivirus software a must. If there is an antivirus software with an outdated VDF, it is as good as having no antivirus software installed, as the new malware will infect the system without getting detected. This method also fails to detect malware that has an ability to change its signature (polymorphic) and the malware that has some portion of its code encrypted.

Virus Signature
A virus signature is a consecutive sequence of bytes that is commonly found in a certain malware sample. That means it's contained within the malware or the infected file and not in unaffected files.

(B) Sandbox detection

In this method, a new application or file is executed in a virtual environment (sandbox) and its behavioural fingerprint is observed for a possible malware. Depending on its behaviour, the antivirus engine determines if it is a potential threat or not and proceeds accordingly. Although this method is a little slow, it is very safe as the new unknown application is not given access to actual resources of the system.

(C) Data mining techniques

This method employs various data mining and machine learning techniques to classify the behaviour of a file as either benign or malicious.

© NCERT not to be republished

Reprint 2026-27

(D) Heuristics

Often, a malware infection follows a certain pattern. Here, the source code of a suspected program is compared to viruses that are already known and are in the heuristic database. If the majority of the source code matches with any code in the heuristic database, the code is flagged as a possible threat.

(E) Real-time protection

Some malware remains dormant or gets activated after some time. Such malware needs to be checked on a real-time basis. In this technique, the anti-malware software keeps running in the background and observes the behavior of an application or file for any suspicious activity while it is being executed i.e. when it resides in the active (main) memory of the computer system.

12.4 Spam

Spam is a broad term and applies to various digital platforms like messaging, forums, chatting, emailing, advertisement, etc. However, the widely recognised form is email spam. Depending on their requirements, organisations or individuals buy or create a mailing list (list of email addresses) and repeatedly send advertisement links and invitation emails to a large number of users. This creates unnecessary junk in the inbox of the receiver's email and often tricks a user into buying something or downloading a paid software or malware.

Nowadays, email services like Gmail, Hotmail, etc. have an automatic spam detection algorithm that filters emails and makes things easier for the end users. A user can also mark an undetected unsolicited email as "spam", thereby ensuring that such type of email is not delivered into the inbox as normal email in future.

12.5 HTTP vs HTTPS

Both the HTTP (Hyper Text Transfer Protocol) and its variant HTTPS (Hyper Text Transfer Protocol Secure) are a set of rules (protocol) that govern how data can be transmitted over the WWW (World Wide Web). In other words, they provide rules for the client web browser and servers to communicate.

HTTP sends information over the network as it is. It does not scramble the data to be transmitted, leaving

Always look for the "https://" at the beginning of the address (URL) of the websites while entering your banking, personal, or other sensitive information.

© NCERT not to be republished

Reprint 2026-27

it vulnerable to attacks from hackers. Hence, HTTP is sufficient for websites with public information sharing like news portals, blogs, etc. However, when it comes to dealing with personal information, banking credentials and passwords, we need to communicate data more securely over the network using HTTPS. HTTPS encrypts the data before transmission. At the receiver end, it decrypts to recover the original data. The HTTPS based websites require SSL Digital Certificate.

Activity 12.1

Ask your teacher to show you how to enable and disable firewall on your computer.

12.6 Firewall

Computer firewall is a network security system designed to protect a trusted private network from unauthorised access or traffic originating from an untrusted outside network (e.g., the Internet or different sections of the same network) to which it is connected (Figure 12.5). Firewall can be implemented in software, hardware or both. As discussed earlier, a malware like worm has the capability to move across the networks and infect other computers. The firewall acts as the first barrier against malware.

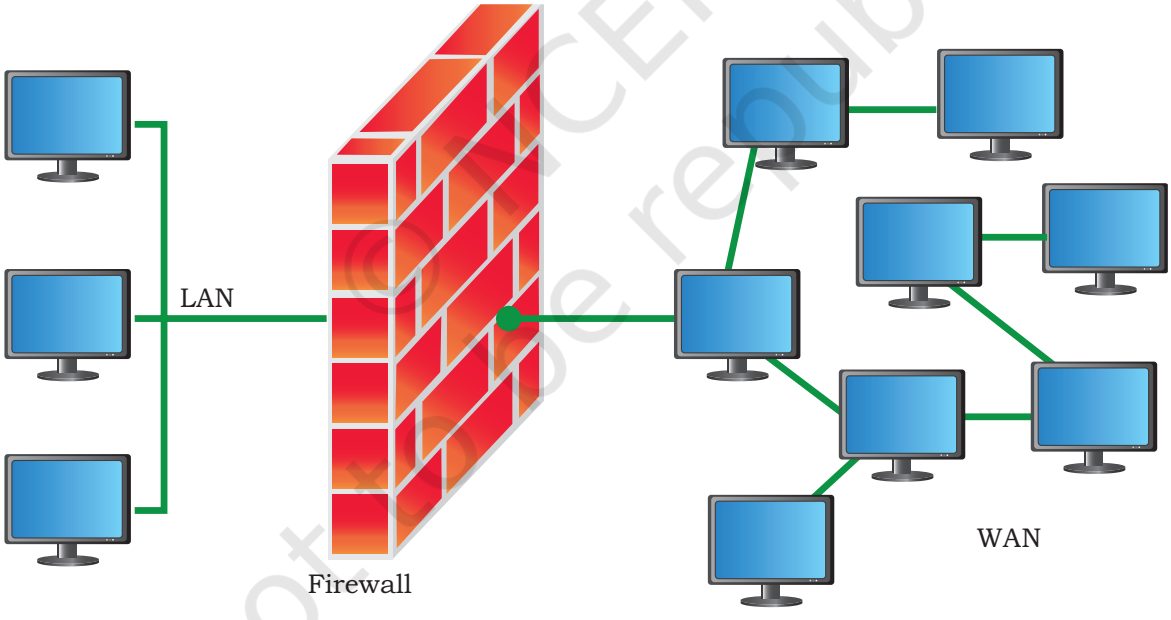


Figure 12.5: A firewall between two networks

A firewall acts as a network filter and based on the predefined security rules, it continuously monitors and controls the incoming and outgoing traffic. As an example, a rule can be set in the firewall of a school LAN, that a student cannot access data from the finance

© NCERT not to be republished

server, while the school accountant can access the finance server.

12.6.1 Types of Firewall

- Network Firewall: If the firewall is placed between two or more networks and monitors the network traffic between different networks, it is termed as Network Firewall.
- Host-based Firewall: If the firewall is placed on a computer and monitors the network traffic to and from that computer, it is called a host-based firewall.

12.7 Cookies

The term "cookie" was derived from the term "magic cookie" used by Unix programmers to indicate a packet of data that a program receives and sends it back unchanged. A computer cookie is a small file or data packet, which is stored by a website on the client's computer. A cookie is edited only by the website that created it, the client's computer acts as a host to store the cookie. Cookies are used by the websites to store browsing information of the user. For example, while going through an e-commerce website, when a user adds items to cart, the website usually uses cookies to record the items in the cart. A cookie can also be used to store other user-centric information like login credentials, language preference, search queries, recently viewed web pages, music choice, favorite cuisine, etc., that helps in enhancing the user experience and making browsing time more productive.

Depending upon their task, there are different types of cookies. Session cookies keep track of the current session and even terminate the session when there is a time-out (banking website). So, if you accidentally left your e-banking page open, it will automatically close after the time-out. Similarly, authentication cookies are used by a website to check if the user is previously logged in (authenticated) or not. This way, you don't need to login again and again while visiting different web pages or links of the same website. You might have also noticed that certain information like your Name, Address, Contact, D.O.B, etc. automatically fills up while filling an online form. This auto-fill feature is also implemented by websites using cookies.

Think and Reflect

Assume students in a class are to finish their project. For this, the access to the Internet has also been given. To ensure maximum output i.e timely completion, can you utilise Firewall to prevent distraction while surfing the net?

Activity 12.2

Open your internet browser and check the settings for cookies. Also, try to locate some cookie files on your computer system.

© NCERT not to be republished

Reprint 2026-27

12.7.1 Threats due to Cookies

Usually, cookies are used for enhancing the user's browsing experience and do not infect your computer with malware. However, some malware might disguise as cookies e.g. "supercookies". There is another type of cookie known as "Zombie cookie" that gets recreated after being deleted. Some third-party cookies might share user data without the consent of the user for advertising or tracking purposes. As a common example, if you search for a particular item using your search engine, a third-party cookie will display advertisements showing similar items on other websites that you visit later. So, one should be careful while granting permission to any websites to create and store cookies on the user computer.

12.8 Hackers and Crackers

Hackers and crackers are people having a thorough knowledge of the computer systems, system software (operating system), computer networks, and programming. They use this knowledge to find loopholes and vulnerabilities in computer systems or computer networks and gain access to unauthorised information. In simple terms, a hacker is a person that is skilled enough to hack or take control of a computer system. Depending on the intent, there are different types of hackers.

A hacktivist is a hacker with an aim to bring about political and social change.

12.8.1 White Hats: Ethical Hacker

If a hacker uses its knowledge to find and help in fixing the security flaws in the system, its termed as White Hat hacker. These are the hackers with good intentions. They are actually security experts. Organisations hire ethical or white hat hackers to check and fix their systems for potential security threats and loopholes. Technically, white hats work against black hats.

12.8.2 Black Hats: Crackers

If hackers use their knowledge unethically to break the law and disrupt security by exploiting the flaws and loopholes in a system, then they are called black hat hackers.

12.8.3 Grey Hats

The distinction between different hackers is not always clear. There exists a grey area in between, which

© NCERT not to be republished

represents the class of hackers that are neutral, they hack systems by exploiting its vulnerabilities, but they don't do so for monetary or political gains. The grey hats take system security as a challenge and just hack systems for the fun of it.

12.9 Network Security Threats

12.9.1 Denial of Service

Denial of Service (DoS) is a scenario, wherein an attacker (Hacker) limits or stops an authorised user to access a service, device, or any such resource by overloading that resource with illegitimate requests. The DoS attack floods the victim resource with traffic, making the resource appear busy. If attackers carry out a DoS attack on a website, they will flood it with a very large number of network packets by using different IP addresses. This way, the web server would be overloaded and will not be able to provide service to a legitimate user. The users will think that the website is not working, causing damage to the victim's organisation. Same way, DoS attacks can be done on resources like email servers, network storage, disrupting connection between two machines or disrupting the state of information (resetting of sessions).

If a DoS attack makes a server crash, the server or resource can be restarted to recover from the attack. However, a flooding attack is difficult to recover from, as there can be some genuine legitimate requests in it as well.

A variant of DoS, known as Distributed Denial of Service (DDoS) is an attack, where the flooded requests come from compromised computer (Zombies) systems distributed across the globe or over a very large area. The attacker installs a malicious software known as Bot on the Zombie machines, which gives it control over these machines. Depending upon the requirement and availability, the attacker activates a network of these Zombie computers known as Bot-Net to carry out the DDoS attack. While as a simple DoS attack may be countered by blocking requests or network packets from a single source, DDoS is very difficult to resolve, as the attack is carried from multiple distributed locations.

© NCERT not to be republished

12.9.2 Intrusion Problems

Network Intrusion refers to any unauthorised activity on a computer network. These activities may involve unauthorised use of network resources (DoS) or threatening the security of the network and the data. Network intrusion is a very serious problem and the network administrator needs to devise strategy and implement various security measures to protect the network. We have already discussed some of the intrusion attacks such as DoS, Trojans, and Worms. The remaining attacks are briefly discussed below.

(A) Asymmetric Routing

The attacker tends to avoid detection by sending the intrusion packets through multiple paths, thereby bypassing the network intrusion sensors.

(B) Buffer Overflow Attacks

In this attack, the attacker overwrites certain memory areas of the computers within the network with code (set of commands) that will be executed later when the buffer overflow (programming error) occurs. Once the malicious code is executed, an attacker can initiate a DoS attack or gain access to the network.

(C) Traffic Flooding

It is one of the most trivial methods of network intrusion. It involves flooding the network intrusion detection system with message packets. This huge load leaves the network detection system incapable of monitoring the packets adequately. The hacker takes advantage of this congested and chaotic network environment to sneak into the system undetected.

12.9.3 Snooping

Snooping means secretly listening to a conversation. In the context of networking, it refers to the process of secret capture and analysis of network traffic. It is a computer program or utility that has a network traffic monitoring capability. In this attack, the hacker taps or listens to a channel of communication by picking all of the traffic passing through it. Once the network packets are analysed by the snooping device or software, it reproduces the exact traffic packets and places them back in the channel, as if nothing has happened. So, if the data that is being sent over the network is not encrypted, it is vulnerable to snooping and eventually

URL Snooping

It is a software package that downloads and stores a web stream as a file, that can be viewed or used later. The common online video downloaders use the same techniques to download videos from the Web.

© NCERT not to be republished

may cause serious damage, depending upon the type of information leak. However, snooping is not always an attack, at times it is also used by network administrators for troubleshooting various network issues. Snooping is also known as Sniffing.

Various snooping software exist that act as network traffic analyser. Besides, various network hubs and switches have a SPAN (Sniffer Port Analyser) port function for snooping.

12.9.4 Eavesdropping

The term eavesdropping has been derived from the literal practice of secretly listening to the conversations of people by standing under the eaves of a house. Unlike snooping, where the network traffic can be stored for later analysis, eavesdropping is an unauthorised real-time interception or monitoring of private communication between two entities over a network. Also, the targets

Figure 12.6: Eavesdropping

are usually the private communication channels like phone calls (VoIP), instant messages, video conference, fax transmission, etc. In older days, eavesdropping was performed on the conventional telephone line and was known as wiretapping. Digital devices like laptops and cell phones that have a built-in microphone or camera can be easily hacked and eavesdropped using rootkit malware.

Eavesdropping is different from Snooping. While the former happens in real time, the latter does not. As an

© NCERT not to be republished

example, in eavesdropping, imagine someone listening to your private conversation with the help of a hidden microphone in your room or by physically standing near the window of your room. However, in snooping, that person may make a copy of a letter that is addressed to your friend and keep the copy with himself and send the original letter to the intended address.

SUMMARY

- Malware is a software developed with an intention to damage computer hardware, software, steal data, or cause any other trouble to a user.
- A virus is a piece of software code created to perform malicious activities and hamper resources of a computer system.
- The Worm is also a malware that incurs unexpected or damaging behaviour on an infected computer system.
- Worms are standalone programs that are capable of working on its own.
- Ransomware is a type of malware that targets user data.
- Ransomware either blocks the user from accessing their own data or threatens to publish their personal data online and demands ransom payment against the same.
- Trojan is a malware, that looks like a legitimate software and once it tricks a user into installing it, it acts pretty much like a virus or a worm.
- Spyware records and sends the collected information to an external entity without the consent or knowledge of a user.
- An adware displays unwanted online advertisements using pop-ups, web pages, or installation screens.
- A keylogger makes logs of daily keyboard usage and may send it to an external entity as well.
- The on-screen keyboard is an application software that uses a fixed QWERTY key layout.
- Online virtual keyboard is a web-based or a standalone software with a randomised key layout every time it is used.
- A malware can take many routes to reach your computer, which include: Downloaded from the

© NCERT not to be republished

Reprint 2026-27

Internet, Spam Email, using infected Removable Storage Devices, and network propagation.

- An antivirus software is used to detect and remove viruses and hence the name anti-virus.
- Antiviruses now come bundled with the prevention, detection, and removal of a wide range of malware.
- Some of the prominent methods of malware identification used by an antivirus include: Signature-based detection, Sandbox detection, Heuristics.
- Any unwanted data, information, email, advertisement, etc. is called Spam.
- HTTP (Hyper Text Transfer Protocol) and HTTPS (Hyper Text Transfer Protocol Secure) are a set of rules or protocol that govern how data can be transmitted over the World Wide Web.
- Firewall is a network security system designed to protect a trusted private network from unauthorised access or traffic originating from an untrusted external network.
- There are two basic types of firewalls — Network Firewall and Host-based Firewall.
- A computer cookie is a small file or data packet, which is stored by a website on the client's computer.
- Cookies are used by the websites to store browsing information of the user.
- Hackers/Crackers find loopholes and vulnerabilities in computer systems or computer networks and gain access to unauthorised information.
- If a hacker uses its knowledge to find and help in fixing the security flaws in the system, its termed as White Hat hacker.
- If hackers use their knowledge unethically to break the law and disrupt security by exploiting the flaws and loopholes in a system, then they are called black hat hackers.
- The grey hats take system security as a challenge and just hack systems for the fun of it.
- The Denial of Service (DoS) attack floods the victim resource with traffic, making the resource appear busy.
- Distributed Denial of Service (DDoS) is an attack, where the flooded requests come from

© NCERT not to be republished

Reprint 2026-27

compromised computer (Zombies) systems distributed across the globe or over a very large area.

- Network Intrusion refers to any unauthorised activity on a computer network.
- Snooping is the process of secret capture and analysis of network traffic by malicious users.
- Eavesdropping is an unauthorised real-time interception or monitoring of private communication between two entities over a network.

EXERCISE

1. Why is a computer considered to be safe if it is not connected to a network or Internet?
2. What is a computer virus? Name some computer viruses that were popular in recent years.
3. How is a computer worm different from a virus?
4. How is Ransomware used to extract money from users?
5. How did a Trojan get its name?
6. How does an adware generate revenue for its creator?
7. Briefly explain two threats that may arise due to a keylogger installed on a computer.
8. How is a Virtual Keyboard safer than On Screen Keyboard?
9. List and briefly explain different modes of malware distribution.
10. List some common signs of malware infection.
11. List some preventive measures against malware infection.
12. Write a short note on different methods of malware identification used by antivirus software.
13. What are the risks associated with HTTP? How can we resolve these risks by using HTTPS?
14. List one advantage and disadvantage of using Cookies.
15. Write a short note on White, Black, and Grey Hat Hackers.
16. Differentiate between DoS and DDoS attack.
17. How is Snooping different from Eavesdropping?

© NCERT not to be republished

Reprint 2026-27

Chapter 13

Project Based Learning

12130CH13

"An idea that is developed and put into action is more important than idea that exists only as an idea."

— *Gautam Buddha*

In this Chapter

- *Introduction*
- *Approaches for Solving Projects*
- *Teamwork*
- *Project Descriptions*

13.1 Introduction

Project based learning gives a thorough practical exposure to students regarding a problem upon which the project is based. Through project based learning, students learn to organise their project and use their time effectively for successful completion of the project. Projects are developed generally in groups where students can learn various skills such as working together, problem solving, decision making, and investigating activities. Project based learning involves the steps such as analysing the problem, formulating the problem into small modules, applying the mechanism or method to solve each module and then integrating the solution of all the modules to arrive at the complete solution of the problem. To solve a problem, it is required that those who work on it gather the relevant data and process it by applying a particular method. Data may

© NCERT not to be republished

be collected as per the requirement of the project in a particular format. All the team members should be associated to accomplish the task. After collecting data, it should be processed to solve the problem. The results should be reported in a predetermined format.

13.2 Approaches for Solving Projects

The approach followed for the development and completion of a project plays a pivotal role in project based learning. There are several approaches to execute a project such as modular approach, top down approach and bottom up approach. A structured or a modular approach to a project means that a project is divided into various manageable modules and each of the modules has a well-defined task to be performed with a set of inputs. This would lead to a set of outputs which when integrated leads to the desired outcome.

Different steps involved in project based learning (Figure 13.1) are :

(1) **Identification of a project:** The project idea may come through any real-life situation. For example, one could think of doing a project for organising a seminar. One needs to understand the usefulness of the project and its impact. Students must be encouraged to undertake interdisciplinary projects.

(2) **Defining a plan:** Normally for any kind of project, there are several project members involved in it. One project leader has to be identified. The roles of project leader and each project member have to be clearly defined. Students who are performing a project must be assigned with specific activities. The various tools for executing these activities must be known. To obtain a better solution, one should always think of the extreme situations.

(3) **Fixing of a time frame and processing:** Every project is a time relevance project. A student must understand the importance of time frame for completion of the project. All the activities which are performed in the projects require a certain amount of time. Every project must be well structured and at the same time it must be flexible in its time frame.

© NCERT not to be republished

Reprint 2026-27

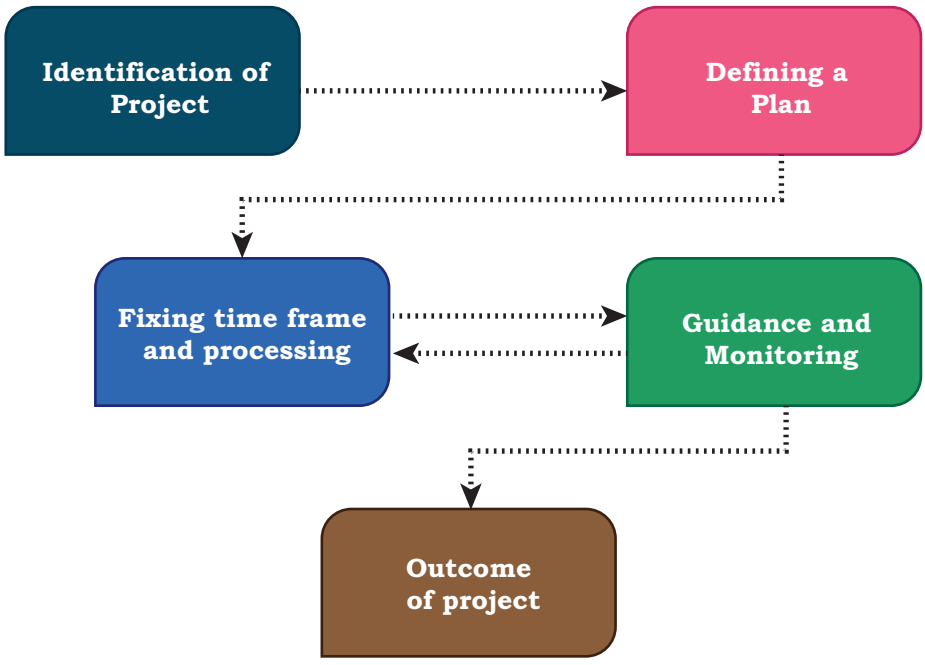


Figure 13.1: Steps in project based learning

(4) **Providing guidance and monitoring a project:** Many times, the participants in the project get stuck up with a particular process and it becomes impossible to proceed further. In such a case, they need guidance, which can be obtained from various resources such as books, websites and experts in the field. While it is essential that the project leader should ensure monitoring of the project, the guide teacher also helps in monitoring the project.

(5) **Outcome of a project:** One needs to understand thoroughly the outcome of a project. The outcome can be single, or it can be multiple. The output of a project can be peer reviewed and can be modified as per the feedback from the guide teacher or other users.

13.3 Teamwork

Many real-life tasks are very complex and require a lot of individuals to contribute in achieving them. Efforts made by individuals collectively to accomplish a task is called teamwork.

For example, in many sports, there is a team of players. These players play together to win a match. Take an example of a cricket team. We find that even if a bowler bowls a good ball but if the fielder cannot take

© NCERT not to be republished

Reprint 2026-27

a catch then the wicket cannot be taken. So, in order to take a catch, efforts of a bowler as well as of fielders are needed. To win a cricket match, contributions from all the team members in all the three areas batting, bowling and fielding are required.

13.3.1 Components of Teamwork

Apart from technical proficiency, a wide variety of other components make a successful teamwork. It comprises skilled team members with specific roles to achieve the goal.

(A) Communicate with Others

When a group of individuals perform one job, it is necessary to have effective communication between the members of the team. Such communication can be done via e-mails, telephones or by arranging group meetings. This helps the team members to understand each other and sort out their problems to achieve the goal effectively.

(B) Listen to Others

It is necessary to understand the ideas of others while executing a job together. This can be achieved when the team members listen to each other in group meetings and follow steps that are agreed upon.

(C) Share with Others

Ideas, images and tools need to be shared with each other in order to perform a job. Sharing is an important component of teamwork. Any member of the team who is well versed in a certain area should share the expertise and experience with others to effectively achieve the goal within the time frame.

(D) Respect for Others

Every member of the team must be treated respectfully. All the thoughts and ideas that are put forth in the group meetings may be respected and duly considered. Not respecting the views of a particular member may cause problems and that particular team member may not give his best.

(E) Help Others

A helping hand from every member is a key to success. Sometimes help from people who are not a part of the team is also obtained in order to accomplish a job.

© NCERT not to be republished

(F) Participate

All the team members must be encouraged by each other to participate in completing the project and also in discussions in group meetings. Also, every member should take an active participation so that they feel their importance in the team.

13.4 PROJECT DESCRIPTIONS

In this section, some examples of project works are given, which can be taken up in groups under project based learning. However, a group may choose any other project in consultation with the guide teacher.

Project Title 1: Automation of Order Processing in a Restaurant

Description

A new restaurant "Stay Healthy" is coming up in your locality. The owner/management of the restaurant wants to use a computer to generate bills and maintain other records of the restaurant. Your team is asked to develop an application software to automate the order placing and associated processes.

Specifications

Make a group of students to undertake a project on automating the order processing of the restaurant 'Stay Healthy'. The owner of the restaurant wants the following specific functionalities to be made available in the developed application:

- There should be two types of Login options — one for the manager of the joint and other for the customer.
- Kiosk(s) running the software for customers will be placed at reception for placing the order. On the opening screen, menu for placing orders will be displayed.
- To place orders, customers will enter Item Code(s) and quantity desired.
- After placing an order, a soft copy of the bill will be displayed on the kiosk, having an Order Number.
- Every bill will have a unique identification (such as combination of date, and order number of the day) and should be saved in the data file/database.
- Order Number starts from 1 every day.

© NCERT not to be republished

Reprint 2026-27

- For Manager login—provision for entry/change of Menu, deletion of Order (on demand) and generation of following report is desired.
  - ✓ A Report giving Summary of the Sales made on a Day. Program should accept the date for which the Summary is required.
- Add at least one more relevant report of your choice to the program.

Project Title 2 : Development of a Puzzle

Description

Implement a puzzle solving game in Python. The game presents a grid board composed of cells to the player, in which some cells have Bomb. Player is required to clear the board (of the bomb), without detonating any one of them with the help of clue(s) provided on the board.

Specifications

For clearing the board, the player will click a cell on the board, if the cell contains a bomb, the game finishes. If the cell does not contain a bomb, then the cell reveals a number giving a clue about the number of bombs hidden in adjacent cells.

Before you start coding the game, play any Minesweeper game five times. This will help you in proper understanding of your project. To reduce the complexity of the program you can fix the grid size to 6x6 and number of bombs to 6.

Note: Do ensure to handle various exception(s) which may occur while playing the game, in your code.

Project Title 3 : Development of an Educational Game

Description

You are a member of the ICT club of your school. As a club member, you are given the responsibility of identifying ways to improve mathematical skills of kids, in the age group of 5-7 years. One of the club members suggested developing an Edutainment Game named "Match the Sum" for it. Match the Sum will hone summing skills of student(s), by allowing them to form number 10 by adding 2/3 digits.

© NCERT not to be republished

Specifications

Following are the details of provisions required for program:

- Display a list of 15 cells on screen, where each cell can hold a digit (1 to 9)
- Randomly generate a digit at a time and place it in the list from the right end. Program will keep on generating digits at equal intervals of time and place it in the rightmost cell. (Already existing digits, will be shifted left, by one cell, with every new addition of digits' in the list)
- For playing the game, students' will be allowed to type 2/3 digits (one at a time) currently displayed in the list of cells.
- If the sum of those digits is 10, then those digits should get removed from the list of cells.
- Game will continue till there is an empty cell to insert a digit in the list of cells.

Note: Do take care of the situation when digits displayed in a list of cells do not add up to 10.

© NCERT not to be republished

© NCERT not to be republished